Study Report 2004-04

Replication of Zeidner, Johnson, and Colleagues' Method for Estimating Army Aptitude Area (AA) Composites

Tirso Diaz
Michael Ingerick
Mary Ann Lightfoot
Human Resources Research Organization

20040917 045

United States Army Research Institute for the Behavioral and Social Sciences

August 2004

U.S. Army Research Institute for the Behavioral and Social Sciences

A Directorate of the U.S. Army Human Resources Command

ZITA M. SIMUTIS
Director

Research accomplished under contract
for the Department of the Army

Human Resources Research Organization

Technical Review by

Peter M. Greenston, U.S. Army Research Institute
Trueman R. Tremble, U.S. Army Research Institute

NOTICES

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy) August 2004	2. REPORT TYPE Final	3. DATES COVERED (from... to) February 2001 - September 2003

4. TITLE AND SUBTITLE Replication of Zeidner, Johnson, and Colleagues' Method for Estimating Army Aptitude Area (AA) Composites	5a. CONTRACT OR GRANT NUMBER DASW01-98-D-0047 / Delivery Order 0030
	5b. PROGRAM ELEMENT NUMBER 665803
6. AUTHOR(S) Tirso Diaz, Michael Ingerick, & Mary Ann Lightfoot (Human Resources Research Organization)	5c. PROJECT NUMBER D730
	5d. TASK NUMBER 263
	5e. WORK UNIT NUMBER C01
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Human Resources Research Organization 66 Canal Center Plaza, Suite 400 Alexandria, VA 22314	8. PERFORMING ORGANIZATION REPORT NUMBER DFR 03-87
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences 2511 Jefferson Davis Highway Arlington, VA 22202-3926	10. MONITOR ACRONYM ARI
	11. MONITOR REPORT NUMBER Study Report 2004-04

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Peter Greenston, Contracting Officer's Technical Representative

14. ABSTRACT *(Maximum 200 words)*:

To select and classify recruits to entry-level jobs, the Army employs nine Aptitude Area (AA) composites that are derived from the Armed Services Vocational Aptitude Battery (ASVAB). Effective January 2002, the Army adopted an "interim" set of nine AA composites based on empirically estimated weights for a seven ASVAB test battery. Zeidner, Johnson, and colleagues had developed these "interim" composites as part of ongoing research into improved military classification systems. The purpose of the present study was to independently replicate and document – as a prerequisite for subsequent evaluation – the Zeidner, Johnson, and colleagues' method, and previously reported results, for the 9, 17, and 150 composites comprising their proposed classification system. Following Zeidner, Johnson, and colleagues' method, the present study successfully reproduced the 9, 17, and 150 composites. These findings support the operational use of the 9 "interim" composites, as well as the use of the 9/17/150 composites in future research and policy analysis evaluating the potential of the proposed two-tiered classification system to substantially improve Army-wide classification and assignment.

15. SUBJECT TERMS

Military Personnel Classification; Army Aptitude Area (AA) Composites; Armed Services Vocational Aptitude Battery (ASVAB)

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGES	21. RESPONSIBLE PERSON (Name and Telephone Number)
16. REPORT	17. ABSTRACT	18. THIS PAGE			
Unclassified	Unclassified	Unclassified	Unlimited	83	Peter Greenston (703) 602-7944; DSN 332-7944

Study Report 2004-04

Replication of Zeidner, Johnson, and Colleagues' Method for Estimating Army Aptitude Area (AA) Composites

Tirso Diaz
Michael Ingerick
Mary Ann Lightfoot
Human Resources Research Organization

Selection and Assignment Research Unit
Michael G. Rumsey, Chief

U.S. Army Research Institute for the Behavioral and Social Sciences
2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926

August 2004

Army Project Number 20465803D730	Personnel and Training Analysis Activities

Approved for public release; distribution is unlimited.

FOREWORD

Assigning tens of thousands of Army recruits per year to job training for which they are best suited and in ways that maximize aggregate Soldier performance represents a major challenge. The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) has a long history of conducting and supporting research aimed at improving the Army's selection and classification process. In that process potential recruits take the Armed Services Vocational Aptitude Battery (ASVAB), from which nine Aptitude Area (AA) composites are calculated for selection and classification of recruits into entry-level jobs. Until recently, each AA composite was built from four (of the nine) ASVAB subtests. In January 2002, the Army adopted fractional weights based on a defensible performance criterion. This allowed all subtests to contribute to all composites in proportion to their power to "explain" Soldier performance. An ARI contractor research team had developed these composites as part of a larger research program into improving military classification systems.

The purpose of the present study was to independently replicate and document the Zeidner, Johnson, and colleagues' method for estimation of the nine (as well as alternative) composites comprising the proposed classification system, as a prerequisite to their subsequent evaluation (reported separately). The present study successfully reproduced the composites previously reported. These findings support the operational use of the nine composites, as well as the alternative composites, in future research and policy analysis aimed at evaluating the potential benefits of the proposed classification system for improving Army-wide classification and assignment.

Barbara A. Black

BARBARA A. BLACK
Acting Technical Director

ACKNOWLEDGEMENTS

The authors thank Dr. Joseph Zeidner, Mr. Cecil Johnson and, in particular, Mr. Yefim Vladimirsky for making available their original computational programs and related data files, and for promptly responding to our inquiries.

REPLICATION OF ZEIDNER, JOHNSON, AND COLLEAGUES' METHOD FOR ESTIMATING ARMY APTITUDE AREA (AA) COMPOSITES

EXECUTIVE SUMMARY

Research Requirement:

To select and classify recruits to job training, the Army employs nine Aptitude Area (AA) composites. Effective January 2002, the Army adopted a set of nine AA composites based on empirically estimated weights. Developed by Zeidner, Johnson, and colleagues (Zeidner, Johnson, Vladimirsky, & Weldon, 2000, 2001), with support from ARI, these composites were part of a proposed two-tiered classification system designed to substantially enhance the classification potential of the Army's AA composites. The purpose of the present study was to independently replicate and document Zeidner, Johnson, and colleagues' method of deriving the composites, in particular the nine AA composites currently in operational use as well as the alternative 17 and 150 composites comprising the two-tiers of the proposed system. Replication was undertaken as a prerequisite to subsequent evaluation of these composites (reported separately).

Method:

The present study replicated Zeidner, Johnson, and colleagues' method using the same Skills Qualification Test (SQT) database (*N* = 257,810) and affiliated data used by Zeidner et al. (2000, 2001) to generate the AA composites. This procedure involved: (1) correcting ASVAB subtest-SQT validities for criterion unreliability and multivariate range restriction by MOS; (2) aggregating corrected ASVAB subtest-SQT validities by job family; (3) using the aggregated ASVAB subtest-SQT validities, empirically estimating population beta weights for each job

family; (4) converting population beta weights (from Step 3) to unstandardized b-weights and constants; and (5) transforming these unstandardized b-weights and constants to operational weights and constants for computing AA standard scores.

Findings:

Overall, we were able to reproduce both population beta weights and operational weights (and constants) for the 9, 17, and 150 composites previously reported by Zeidner et al. (2000, 2001) with one exception. The single exception was associated with the Clerical Administration 1 (CL1) composite in the 17 job family configuration comprising Zeidner, Johnson, and colleagues' second tier. This difference owes to Zeidner et al.'s (2000, 2001) particular implementation of the recommended procedure for deriving the best set of positive weights.

Use of Findings:

Our results validate the Zeidner, Johnson, and colleagues' method for deriving AA composite weights, including the AA composite weights currently in operational use by the Army. The present study represents an independent verification of their method and previously reported results. These findings support the use of these composites in future research and policy analysis evaluating the proposed two-tiered classification system.

REPLICATION OF ZEIDNER, JOHNSON, AND COLLEAGUES' METHOD FOR ESTIMATING ARMY APTITUDE AREA (AA) COMPOSITES

CONTENTS

INTRODUCTION

Background

To select and classify recruits to job training, the Army employs nine Aptitude Area (AA) composites. Each composite represents a differentially weighted function of aptitudes and skills required for successful performance. Until recently, these nine AA composites utilized unit-weights (i.e., 0, 1). Based on rational linkages to job content, these unit weights were meant to reflect the relative importance of different cognitive aptitudes and abilities (e.g., verbal ability, coding speed, mechanical comprehension), as measured by the Armed Services Vocational Battery (ASVAB), in determining job performance within a family of jobs (e.g., Clerical, Combat, Field Artillery, etc.). These "original", unit-weighted AA composites and corresponding ASVAB subtests are shown in Table 1.

Table 1: ASVAB Subtests Comprising the Army's "Original" AA Composites and AFQT

	ASVAB SUBTESTS								
	AR	MK	VE	AS	EI	GS	MC	CS	NO
AA COMPOSITES									
Electronics Repair	X	X			X	X			
General Maintenance		X		X	X	X			
Mechanical Maintenance				X	X		X		X
Operators / Food			X	X			X		X
Surveillance / Communications	X		X	X			X		
Combat	X			X			X	X	
Field Artillery	X	X					X	X	
Skilled Technical		X	X			X	X		
Clerical	X	X	X						
General Technical	X		X						
AFQT	X	X	XX						

ASVAB is comprised of following subtests: Arithmetic Reasoning (AR), Math Knowledge (MK), Verbal (VE) = Paragraph Comprehension (PC) + Word Knowledge (WK), Auto & Shop Information (AS), Electronics Information (EI), General Science (GS), Mechanical Comprehension (MC), Coding Speed (CS), Numerical Operations (NO).

Starting in January 2002, the Army adopted a set of nine AA composites based on empirically estimated beta weights, corrected to the Youth population, for a 7 ASVAB test battery (Greenston, Rumsey, Zeidner, & Johnson, 2001).[1] The ASVAB subtest weights that define the AA composites are shown in Table 2. These composites were developed by

Table 2: ASVAB Subtest (Relative) Weights Comprising the AA Composites

	ASVAB Subtests						
	AR	MK	VE	AS	EI	GS	MC
AA COMPOSITES							
Electronics Repair	.818	.890	1.000	.754	.598	.151	.469
General Maintenance	.828	.794	.417	1.000	.577	.411	.503
Mechanical Maintenance	.339	.289	.237	1.000	.340	.060	.394
Operators / Food	.962	.600	.714	1.000	.377	.251	.636
Surveillance / Communications	.685	1.000	.915	.437	.551	.019	.386
Combat	.532	1.000	.529	.733	.343	.313	.595
Field Artillery	.715	1.000	.586	.673	.297	.249	.700
Skilled Technical	.727	.697	1.000	.357	.230	.187	.446
Clerical	1.000	.767	.980	.110	.110	.000	.148

Zeidner, Johnson, and colleagues (Zeidner, Johnson, Vladimirsky, & Weldon, 2000, 2001), with support from the Army Research Institute (ARI), as part of a proposed two-tiered classification system.[2] Within this system, first tier composites are intended for classifying recruits to one of 150 entry-level job families. The second tier composites, aimed at a smaller set of job families (9 or 17), are meant for recruiting, vocational counseling, and administration purposes. A program of research conducted by Zeidner, Johnson, and colleagues using large-scale simulations demonstrated that the proposed two-tiered classification system and related

---

[1] DoD initiated a design change to reduce the scope of the ASVAB, from nine to seven subtests, that was implemented in January 2002. Numerical Operations (NO) and Coding Speed (CS) were deleted from the battery, for the purpose of facilitating uniform administration of the test battery, but at a significant reduction in potential classification efficiency.

[2] In descriptions of the Zeidner, Johnson, and colleagues' method, these weights are frequently referred to as least squares estimates (LSE) or LSE weights, as the weights are empirically estimated using conventional ordinary least-squares (OLS) regression.

composites would produce substantial gains in aggregate Soldier performance over the Army's previous system of unit-weighted composites (Johnson, Zeidner, & Leaman, 1992; Statman, 1993; Zeidner et al., 2000, 2001).

The purpose of the present study is to independently replicate and document the Zeidner, Johnson, and colleagues' method of deriving the composites for their proposed two-tiered classification system, in particular the nine AA composites currently in operational use[3], as a prerequisite for subsequent evaluation of the system (reported separately). More specifically, the present study seeks to replicate their method and previously reported results (Zeidner et al., 2000, 2001) for the 9, 17, and 150 composites comprising the two-tiers of this proposed system. For all composites, this includes both the empirically estimated population beta weights and the operational weights (and constants) for computing AA standard scores used by the Army when making personnel and training decisions.[4] Our replication of the Zeidner, Johnson, and colleagues' method is based on technical reports describing their method (Greenston et al., 2001; Zeidner et al., 2001, 2003a, 2003b), supplemented with information contained in their original programs. All SAS programs, with documentation, used in our replication can be found in Appendix A. The original programs implementing the Zeidner, Johnson, and colleagues' method, also with documentation, can be found in Appendix B.

---

[3] Given the importance of the change from unit-weighted to LSE composites, ARI undertook an independent study to confirm the results of the original research and to provide a well-documented record of the methodology.
[4] Zeidner, Johnson, and colleagues refer to these operational weights as "u, k values" or "transformation weights", as they reflect linear transformations of the population beta weights to weights that standardize AA scores to have a mean of 100 and a standard deviation (SD) of 20.

METHOD

Data

The present study employs the same Skills Qualification Test (SQT) program database used by Zeidner, Johnson, and colleagues (Zeidner et al., 2000, 2001, 2003a, 2003b). This database contains ASVAB subtest scores and standardized SQT scores for FYs 1987-1989 ($N =$ 257,810).[5] Running from 1983 to 1991, the SQT was a comprehensive program for assessing enlisted Soldiers' job proficiency for purposes of advancement and promotion. Under this program, Soldiers were required to take the SQT annually after completing 11 months or more of service. SQT were work samples or paper-and-pencil job knowledge tests. Each SQT was specific to a military occupational specialty (MOS). These data were originally collected and made available by ARI. Basic descriptives, and the ASVAB variance-covariance matrix, for the full sample are reported in Appendix C.

Procedure

As an overview, the Zeidner, Johnson, and colleagues method consists of the following steps:

- Step 1: Correct ASVAB subtest-SQT validities for criterion unreliability and multivariate range restriction by MOS.
- Step 2: Aggregate ASVAB subtest-SQT validities by job family.
- Step 3: Using the aggregated ASVAB subtest-SQT validities, empirically estimate population beta weights for each job family.

---

[5] To ensure comparability across MOS, SQT scores were standardized to have a mean of 0 and an SD of 1.

- Steps 4 and 5: Transform population beta weights (from Step 3) to operational weights (and constants) for computing AA standard scores.

Our goal was to replicate the above steps, and their implementation, so as to reproduce the composite validities and weights previously reported by Zeidner, Johnson, and colleagues (Zeidner et al, 2000, 2001, 2003a). Technical details related to each step, and our replication, are summarized in turn.

Step 1: Correct ASVAB Subtest-SQT Validities for Criterion Unreliability and Multivariate Range Restriction by MOS

Consistent with the current psychometric literature (e.g., Cohen, Cohen, West, & Aiken, 2002; Guion, 1998; Hunter & Schmidt, 1990; Ree, Carretta, Earles, & Albert, 1994; Sackett & Yang, 2000; Schmidt & Hunter, 1996), the goal of this step is to correct observed validities for statistical artifacts (criterion unreliability, range restriction) that downwardly bias validities and related estimates (e.g., beta weights). In keeping with recommendations regarding the order of these corrections (Hunter, Schmidt, & Le, 2002; Stauffer & Mendoza, 2001), criterion reliability was corrected first, followed by range restriction. Using a variation of the standard correction for attenuation formula,[6]

$$\rho_{xy} = \frac{r_{xy}}{\sqrt{r_{yy}}} , \qquad (1)$$

observed ASVAB-SQT validities (r_{xy}) were corrected by MOS for criterion unreliability (r_{yy}). Criterion reliabilities used in these corrections reflect internal consistency reliabilities, specifically coefficient alphas (Cronbach, 1951). The reliabilities employed in our corrections were the same as those used by Zeidner et al. (2000, 2001) and are reported in Appendix D.

After correcting for criterion unreliability, ASVAB-SQT validities were corrected for multivariate range restriction by MOS using formulas originally developed by Aitken (1934) and Lawley (1943), and described by Gulliksen (1950), and Birnbaum, Paulson, and Andrews (1950). Large-scale simulations demonstrate that, when applied appropriately, the Aitken-Lawley corrections consistently produce estimates that closely approximate validities for the relevant reference population (Sackett & Yang, 2000). The Aitken-Lawley multivariate range restriction correction formulas are most appropriate for this case because selection to an MOS results from multiple variables (e.g., ASVAB subtests) (Ree et al., 1994; Sackett & Young, 2000).

The multivariate range restriction correction formulas were applied to the validities twice to obtain two separate sets of corrected ASVAB subtest-SQT validities. First, to produce ASVAB subtest-SQT validities corrected to the Army Input population, and the second time, to produce validities corrected to the Youth population. The two sets of corrections were necessary as the relevant reference population, and thereby the factors (or variables) restricting the validities, differs for the two tiers in Zeidner, Johnson, and colleagues' proposed classification system. For purposes of the first tier, the relevant reference population is the Army Input population, so observed ASVAB subtest-SQT validities are restricted by formal classification effects and eligible recruits' self-selection to an MOS. For the second tier, the relevant reference population is the Youth population, so observed validities are doubly restricted by both formal selection (e.g., AFQT) and classification effects, as well as prospective recruits' self-selection into the Army.

---

[6] Unless otherwise specified, notation used throughout this report follows that of Cohen et al. (2003).

Corrections were made to obtain the variance-covariance (VCV) matrix (v) for the relevant reference population shown below (using a variation of the conventional Birnbaum, Paulson, & Andrews [1950] notation, cf., Ree et al. [1994]; Sackett & Yang [2000]):

$$v = \begin{bmatrix} v_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix}, \qquad (2)$$

where v_{xx} is known and denotes the unrestricted (or population) VCV matrix for all 9 ASVAB subtests; v_{xy} (or its transpose, v_{yx}) is unknown and denotes the estimate of the corrected ASVAB subtest-SQT covariances in the relevant reference population; and v_{yy} is unknown and denotes the estimate of the corrected SQT variance in the relevant reference population. To obtain the corrected ASVAB subtest-SQT validities, we did the following. First, using the known unrestricted (or population) variances for all 9 ASVAB subtests (including NO and CS) and the known ASVAB variances and ASVAB subtest-SQT covariances (corrected for criterion unreliability) from a restricted MOS sample, we estimated the ASVAB-SQT covariances corrected to the relevant reference population (Army Input or Youth).[7] Second, we derived the SQT variances for each MOS corrected to the relevant reference population. At this point, we had the complete variance-covariance matrix corrected to the relevant reference population for all 155 MOS. Third, and finally, we converted the corrected variance-covariance matrices for each MOS to ASVAB subtest-SQT validities, reflecting the range-restricted corrected validities for the current operational ASVAB battery of 7 subtests (dropping NO and CS).

It should be noted that including information on the full 9 ASVAB subtests in the correction procedure, even though the current battery only contains 7 (minus NO and CS), is necessary to ensure that the range-restriction corrected validities are accurate. This is because

NO and CS are involved in restricting the variance of SQT scores in the MOS samples, which thereby lowers the observed ASVAB subtest-SQT validities. In order to accurately recover the unknown population (Army Input or Youth) variance of SQT prior to the selection and classification effects underlying the MOS samples, NO and CS need to be accounted for, as done in the Zeidner, Johnson, and colleagues' procedure.

After applying the above correction procedures, we had two sets of ASVAB subtest-SQT validities corrected for criterion unreliability and range restriction for 155 MOS; one set reflecting validities corrected to the Army Input population, and the second set reflecting validities corrected to the Youth population. As an interim step, we verified our results against the corrected validities reported by Zeidner, Johnson, and colleagues (2003a).

Step 2: Aggregate ASVAB Subtest-SQT Validities by Job Family

The goal of this step is to aggregate the corrected ASVAB subtest-SQT validities (from Step 1) by job family for empirically estimating the population beta weights derived in Step 3. The procedure for aggregating these corrected validities to form a weighted average for each job family can be summarized in the formula below:

$$\overline{\rho}_{xy} = \frac{\sum(\rho_{xy} n)}{\sum n} \qquad (3)$$

where n represents the acquisition number for each MOS; that is, the number of Army recruits to be allocated to an MOS. As evident from the formula, this procedure is directly analogous to conventional meta-analytic procedures in applied psychology for aggregating validities across research samples (e.g., Hunter & Schmidt, 1990). Likewise, the procedure is comparable to the

[7] In Zeidner, Johnson, and colleagues' description of their method (Zeidner et al., 2001, 2003a, 2003b), this is referred to as the Youth Population criterion variance (YPCV).

traditional method in statistics of pooling averages across multiple samples. A conceptual description of this aggregation procedure follows.

For each MOS, we multiplied the corrected ASVAB subtest-SQT validities (the ρs from Step 1) separately by their respective acquisition numbers (n). The resulting products (of the SQT validities and the acquisition numbers) for each ASVAB subtest were then summed across all MOS corresponding to a particular job family, as were the applicable acquisition numbers. To obtain the final aggregated job family-level validities, we then divided the sum of the n-weighted ASVAB-SQT validities for a given job family ($\sum \rho_{xy} n$), by the respective sum of MOS acquisition numbers ($\sum n$).

The acquisition numbers (n) used in our replication are exactly the same as the numbers employed by Zeidner, Johnson, and colleagues. These numbers, which are reported in Appendix E, were based on available assignment and classification data from the Seabrook Reports (1989). Multiplying MOS validities by their respective acquisitions numbers ensures that the contribution of an MOS to the estimation of job family-level beta weights accurately reflects its operational importance to Army assignment and classification policy. The job family configurations used in our replication match the configurations proposed by Zeidner, Johnson, and colleagues, for the new two-tiered classification system, which were previously reported in Zeidner et al. (2001).

Upon completing this step, we had obtained the aggregated corrected ASVAB subtest-SQT validities ($\overline{\rho} s$) necessary for empirically estimating the population beta weights (βs) in Step 3. We repeated this procedure three times, to derive three different sets of aggregated ASVAB subtest-SQT validities: (a) one set corresponding to Zeidner, Johnson, and colleagues'

first tier; and (b) two sets corresponding to the second tier (one for 9 job families and a second version for 17 job families).

Step 3: Using the Aggregated ASVAB Subtest-SQT Validities, Empirically Estimate Population Beta Weights for Each Job Family

The primary goal of this step is to empirically estimate population beta weights (βs) for each job family based on the aggregated ASVAB subtest-SQT validities ($\overline{\rho s}$) from the preceding step. These weights become the basis for the operational weights (and constants) subsequently derived in Steps 4 and 5, which are used by the Army to compute AA standard scores to assign entry-level recruits to an MOS, or for vocational counseling purposes and re-assignment of currently enlisted Soldiers. The current step was motivated by a series of large-scale simulations conducted by Zeidner, Johnson, and colleagues demonstrating greater classification efficiency and aggregate Soldier performance using AA composites based on empirically estimated beta weights versus the existing unit-weighted composites (Zeidner et al., 2000, 2001). The procedure for estimating the population beta weights slightly differs by tier.

For the first tier, comprised of 150 job families, population beta weights (βs) were estimated for each job family using: (a) the applicable aggregated ASVAB subtest-SQT validities (from Step 2); and (b) the ASVAB subtest intercorrelation matrix for the relevant reference population, the Army Input population. This procedure is represented in the following formula:

$$\underline{B_{yx}} = R_{xx}^{-1} \underline{R_{xy}}^{T} \tag{4}$$

For each job family, a 7 x 1 vector of population beta weights ($\underline{B_{yx}}$) was generated empirically using standard ordinary least squares (OLS) regression applied to a 7 x 7 matrix of ASVAB

intercorrelations (R_{xx}), and a 7 x 1 vector of ASVAB subtest-SQT validities ($\underline{R_{xy}}$). At the end of this estimation process, we had 150 sets of population beta weights (βs), one set per job family. Conceptually, these weights reflected the relative, unique contribution of different cognitive aptitudes and abilities, as measured by the ASVAB subtests, to job performance for a family of comparable jobs.

For the second tier, comprised of either 9 or 17 job families, population beta weights were estimated for each job family using: (a) the applicable aggregated ASVAB subtest-SQT validities (from Step 2); and (b) the ASVAB subtest intercorrelation matrix for the relevant reference population, in this case the Youth population. Consistent with Zeidner, Johnson, and colleagues, ASVAB subtest intercorrelations were based on normative information for the 1980 Youth population. In contrast to the estimation process for the first tier, population beta weights were constrained to be positive. This was done for operational purposes, so that poor performance on a test, particularly due to low motivation or deliberate distortion, would not significantly contribute to and, thereby, bias assignment and classification decisions. Otherwise, consistent with the first tier, population beta weights were estimated using standard OLS regression.

To derive the "best" set of positive beta weights, where "best" is defined as the subset of positive beta weights yielding the highest multiple R, we did the following. Starting with all 7 ASVAB subtests, then iterating through successively smaller composites consisting of (m-1) tests, we estimated betas for all possible subsets. That is, we estimated betas for all possible subtests of 7, then 6, then 5, then 4, etc., ASVAB subtests. The stopping rule was reached when we had identified the "best" set of positive beta weights. Specifically, these are the beta weights of the non-negatively weighted composite with the largest possible number of tests, say m^*,

whose multiple R is highest among *all non-negatively* weighted composites with m^* tests and, at the same time, higher than the Rs of *all possible* composites with (m^*-1) tests. Once identified, this set of betas was outputted, with all excluded tests being assigned beta weights of 0.

We carried out this estimation process twice, once for the 9 job families and once for the 17 job families comprising Zeidner, Johnson, and colleagues, second tier. At the end of this estimation process, we had two full sets of population beta weights.

All population beta weights for the 9, 17, and 150 job families estimated from our replication are reported in Appendix F.

Steps 4 and 5: Transform Population Beta Weights (From Step 3) to Operational Weights (and Constants) for Computing AA Standard Scores

The goal for the final two steps is to transform population beta weights (from Step 3) to operational weights (and constants) for purposes of computing AA standard scores for use by Army personnel managers when making classification and counseling decisions.[8] For the first tier, the operational weights are simply unstandardized population b-weights and constants. Consistent with their intended use, the operational weights for the second tier are linearly transformed b-weights and constants, constructed so as to produce AA standard scores with a mean of 100 and a standard deviation (SD) of 20, which reflect the desired population-level characteristics of the relevant reference population, the Youth population. The Zeidner, Johnson, and colleagues' transformation process, which we replicated, generally involves two steps.

[8] Readers are reminded that Zeidner, Johnson, and colleagues refer to these operational weights (and constants) as "u, k values" or "transformation weights".

In the first step, population beta weights (the βs from Step 3) are converted to unstandardized b-weights and constants. For each job family, the population beta weights were converted to unstandardized population b-weights (B_{yx}^*) using the following formula:

$$B_{yx}^* = \beta_{yx}\left(\frac{\sigma_y}{\sigma_x}\right) \tag{5}$$

As can be seen from the formula, this conversion was carried out by multiplying each population beta weight (β_{yx}) by the ratio of the SQT standard deviation over the respective ASVAB subtest standard deviation $\left(\frac{\sigma_y}{\sigma_x}\right)$. For the first tier, the constants (B_{0y}^*) were then derived using the formula:

$$B_{0y}^* = \sum (B_{yx}^* \ast \mu_x) \tag{6}$$

where the set of unstandardized b-weights (B_{yx}^*) for a given job family were individually multiplied by their respective ASVAB subtest population means (μ_x, Army Input), and the summation is across the 7 ASVAB subtests. This step effectively converts the population beta weights back to their original metric (or scale). For the first tier, no additional conversion was necessary, as the unstandardized b-weights and constants produced at this step constitute the operational weights. For the second tier, the conversion to unstandardized b-weights permits the subsequent linear transformation of these same weights (and constants) to an alternative standard scale with the aforementioned properties (mean of 100, SD of 20).

In the second step, applicable only to the second tier, the unstandardized b-weights for each job family were transformed to their final operational form using the formulas:

$$CM = \frac{20}{(10\sqrt{\underline{B_{yx}}^* R_{xx} \underline{B_{yx}}^{*T}})} \tag{7}$$

$$b_{yx} = CM\,(B_{yx}^*) \tag{8}$$

$$b_{0y} = 100 - \sum (b_{yx} * 50) \quad (9)$$

The first formula (Eq. 7) generates a composite multiplier (CM), which standardizes b-weighted AA scores to have a SD of 20. This standardization is accomplished through formula (Eq. 8), which linearly transforms the unstandardized b-weights to operational weights using *CM*. The third and final formula (Eq. 9) derives constants that center operational AA scores to have the desired mean of 100. At the end of this second step, we had the operational weights and intercepts (by job family) for both the 9 and 17 composites comprising Zeidner, Johnson, and colleagues' proposed second tier.

Our replicated operational weights and intercepts for 9, 17, and 150 composites are reported in Appendix F.

RESULTS

Following the Zeidner, Johnson, and colleagues' method, we were able to successfully reproduce both population beta weights and operational weights and intercepts for the 9, 17, and 150 composites previously reported by Zeidner et al. (2000, 2001) with one exception (see Appendix F for our results). The exception can be found in the 17 job family variation of Zeidner, Johnson, and colleagues' second tier, specifically with the Clerical Administration 1 (CL1) composite (see Tables 4 and 5, Appendix F).

Population beta weights reported for CL1 (Zeidner et al., 2000, 2001) were not consistent with the weights generated by our replication. The most notable discrepancies between the Zeidner et al. estimates (2001) and those produced from our replication are associated with the Arithmetic Reasoning (AR) and Mathematical Knowledge (MK) subtests. Specifically, our estimated population beta weights for AR and MK were .24818270 and .22718282, respectively; whereas Zeidner et al. (2000, 2001) reported betas of .00000 (AR) and .36801 (MK). Because

there were differences with the population beta weights, the operational weights previously reported for CL1 were likewise not in agreement with the weights derived from our replication.

An examination of the regression of all possible subsets of ASVAB subtests indicates that the set originally derived by Zeidner et al. (2000, 2001) yielded a multiple R of .5085 based on 4 subtests, compared to our set of best positive weights, which yielded a higher multiple R of .5241 based on 3 subtests.[9] This finding suggests that for this particular composite, the previously reported results are not entirely consistent with the stated goal of the Zeidner, Johnson, and colleagues' (2000, 2001) method for deriving the "best" set of positive weights. The Zeidner et al. results could be viewed as advantageous in that the set of weights generated are based on a larger number of ASVAB subtests, although our weights are consistent with the goal of estimating the "best" set of positive weights and produced a higher multiple R. Additionally, there is a meaningful relationship between our weights for CL1 and the weights estimated for CL (Clerical Administration) at the 9-composite level, such that averaging the weights associated with CL1 and CL2 recover the weights estimated for the CL composite.

Other than the CL1 composite, all other replicated beta weights and operational weights (and constants) comprising the first and second tier of Zeidner, Johnson, and colleagues' proposed classification system matched exactly with previously reported estimates.

DISCUSSION

Starting in January 2002, the Army adopted a set of nine AA composites based on empirically estimated beta weights, corrected to the Youth population, for a 7 ASVAB test battery. Zeidner, Johnson, and colleagues (Zeidner et al., 2000, 2001) developed these

---

[9] More specifically, Zeidner et al. derived the best *positive* weights based on four subtests, whereas our replication of their method produced the best weights *among all possible subsets* based on 3 subtests.

"interim" composites as part of a proposed two-tiered classification system. The purpose of the present study was to independently replicate the Zeidner, Johnson, and colleagues' method, and implementation of their method, in deriving the 9, 17, and 150 composites comprising their proposed classification system.

Overall, we successfully reproduced both population beta weights and operational weights and constants for the 9, 17, and 150 composites previously reported by Zeidner et al. (2000, 2001) with one exception. The single exception is associated with the Clerical Administration 1 (CL1) composite in the 17 job family configuration of Zeidner, Johnson, and colleagues' second tier. Otherwise, we fully replicated population beta weights and operational weights (and constants) for all other composites comprising both the first and second tier of Zeidner, Johnson, and colleagues' proposed classification system, including the "interim" set of nine AA composites currently in operational use by the Army. These findings support the use of these composites in future research and policy analysis aimed at evaluating the potential benefits of the proposed two-tiered classification system for improving Army-wide classification and assignment.

REFERENCES

Aitken, A.C. (1934). Note on selection from a multivariate normal population. *Proceedings of the Edinburgh Mathematical Society*, *4*, 106-110.

Birnbaum, Z.W., Paulson, E., & Andrews, F.C. (1950). On the effect of selection performed on some coordinates of a multi-dimensional population. *Psychometrika*, *15*, 191-204.

Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*, 297-334.

Greenston, P., Rumsey, M., Zeidner, J., & Johnson, C.D. (2001, September 7). *U.S. Army Research Institute classification research and the development of new composites.* Presentation made to Expert Review Panel, Alexandria, VA.

Guion, R.M. (1998). *Assessment, measurement, and prediction for personnel decisions.* Mahwah, NJ: Lawrence Erlbaum Associates.

Gulliksen, H. (1950). *Theory of mental tests*. New York: Wiley.

Hunter, J.F., & Schmidt, F.L. (1990). *Methods of meta-analysis: Correcting error and bias in research findings*. Newbury Park, CA: Sage Publications.

Hunter, J.F., Schmidt, F.L., & Le, H. (2002). *Implications for direct and indirect range restriction for meta-analysis methods and findings*. Manuscript submitted for publication.

Lawley, D.N. (1943). A note on Karl Pearson's selection formulae. *Proceedings of the Royal Society of Edinburgh*, *62* (Section A, Pt. I), 28-30.

Pedhazur, E.J. (1997). *Multiple regression in behavioral research: Explanation and prediction* (3rd ed.). New York: Harcourt Brace.

Ree, M.J., Carretta, T.R., Earles, J.A., & Albert, W. (1994). Sign changes when correcting for range restriction: A note on Pearson's and Lawley's selection formulas. *Journal of Applied Psychology, 79*, 298-301.

Sackett, P.R., & Yang, H. (2000). Correction for range restriction: An expanded typology. *Journal of Applied Psychology, 85*, 112-118.

Schmidt, F.L., & Hunter, J.F. (1996). Measurement error in psychological research: Lessons from 26 research scenarios. *Psychological Methods, 1*, 199-223.

Stauffer, J.M., & Mendoza, J.L. (2001). The proper sequence for correcting correlation coefficients for range restriction and unreliability. *Psychometrika, 66*, 63-68.

Zeidner, J., Johnson, C.D., Vladimirsky, Y., & Weldon, S. (2000). *Specifications for an operational two-tiered classification system for the Army, Volume 1* (TR-1108-VOL-1). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Zeidner, J., Johnson, C.D., Vladimirsky, Y., & Weldon, S. (2001). *Reducing the number of tests in the Armed Services Vocational Aptitude Battery (ASVAB)* (SN-2001-01). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Zeidner, J., Johnson, C.D., Vladimirsky, Y., & Weldon, S. (2003a). *Determining composite validity coefficients for Army jobs and job families* (SN-2003-02). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Zeidner, J., Johnson, C.D., Vladimirsky, Y., & Weldon, S. (2003b). *Determining mean predicted performance for Army job families* (SN-2003-03). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

APPENDIX A: PROGRAMS FOR REPLICATING ZEIDNER, JOHNSON, AND COLLEAGUES' (2000) METHOD

Step 1: Correction for Criterion Unreliability and Range Restriction

Correction for Criterion Unreliability

```
/******************************************************************************
 Correcting ASVAB-SQT validity coefficients and covariances
 for criterion unreliability
******************************************************************************/

libname DirPath "D:\NewAA\";

/****************************************************************
 OUTPUT DATASETS - rename right-hand-side as needed
****************************************************************/
%let Ryx_CorrectRelib_MOS=DirPath.CorXY;
%let Cyx_CorrectRelib_MOS=DirPath.CorCV;

/****************************************************************
 INPUT DATASETS
****************************************************************/
%let Descriptive_MOS=DirPath.Descrip;
%let R_Samp_MOS=DirPath.Corr;

/* variable names for ASVAB subtests                              */
%let TESTNAMES=GS AR NO CS AS MK MC EI VE;


proc iml;

  TestNames = {&TESTNAMES};

  /* Read ASVAB-SQT correlations and numeric ID into XYcorr and MOSNumID */
  use &R_Samp_MOS;
  read all var(TestNames) where(NAMES="SQT") into XYcorr;
  read all var{MOSNUMID} where(NAMES="SQT") into MOSNumID;
  close &R_Samp_MOS;

  /* Read SQT reliabilities into vector YYscal*/
  use DirPath.Relib;
  read all var{YY} into YYscal;
  close DirPath.Relib;

  NumMOS = nrow(MosNumID);    /* =nrow(CorMat) */
  NumTest =ncol(TestNames);

  /* correcting validity coefficients for attenuation using standard formula */
  correctedRxy = XYcorr#(1/SQRT(YYscal));


  /* initialize corrected covariance matrix */
  correctedCVxy = repeat(0,NumMOS,NumTest);

  /* computing corrected covariances -- one ASVAB subtest column at a time */
  use &Descriptives_MOS;
  read all var{SD} where(names="SQT") into sdY;
  do iTest = 1 to NumTest;
    xName = TestNames[iTest];
    read all var{SD} where(NAMES=xName) into sdX;
    correctedCVxy[,iTest] = correctedRxy[,iTest]#sdY#sdX;
```

```
  end;
  close &Descriptives_MOS;

  /* creating dataset of MOS corrected SQT-ASVAB correlations */
  create &Ryx_CorrectRelib_MOS var{MOSNUMID &TESTNAMES};
  correctedRxy = MosNumID || correctedRxy;
  append from correctedRxy;
  close &Ryx_CorrectRelib_MOS;

  /* creating dataset of MOS corrected SQT-ASVAB covariances */
  create &Cyx_CorrectRelib_MOS var{MOSNUMID &TESTNAMES};
  correctedCVxy = MosNumID || correctedCVxy;
  append from correctedCVxy;
  close &Cyx_CorrectRelib_MOS;

quit;
run;
```

Correction for Restriction in Range

```
/******************************************************************************
 Correcting ASVAB-SQT validity coefficients and covariances for
 range restriction.
******************************************************************************/

libname DirPath "D:\NewAA\";

/*************************************************************
 OUTPUT DATASETS - rename right-hand-side as needed
*************************************************************/

/* uncomment line below if correcting to Army input population */
%let Ryx_CorrectRelibRange_MOS=DirPath.ArmyValid;

/* uncomment line below if correcting to Youth population      */
* %let Ryx_CorrectRelibRange_MOS=DirPath.YouthValid_range;


/*************************************************************
 INPUT DATASETS
*************************************************************/
/* uncomment line below if correcting to Army input population */
%let RefPopStats=DirPath.SummaryStats_Army;

/* uncomment line below if correcting to Youth population      */
*%let RefPopStats=DirPath.PopCovYouth;

/* all data sets below common to both reference populations    */
%let Descriptive_MOS=DirPath.Descrip;
%let R_Samp_MOS=DirPath.Corr;
%let C_Samp_MOS=DirPath.Covar;
%let Ryx_CorrectRelib_MOS=DirPath.CorXY;
%let Cyx_CorrectRelib_MOS=DirPath.CorCV;

/* variable names for ASVAB subtests                           */
%let TESTNAMES=GS AR NO CS AS MK MC EI VE;


proc iml;

  TestNames = {&TESTNAMES};

  /* read REFERENCE POPULATION ASVAB subtests covariance */
  use &RefPopStats;
  read all var{&TESTNAMES} where(_TYPE_='COV' & names?TestNames) into PopCxx;
  close &RefPopStats;
  SDx = sqrt(vecdiag(PopCxx));

  /* open MOS SQT-ASVAB covariance corrected for unreliability */
  use &Cyx_CorrectRelib_MOS;
  read all var{MOSNUMID} into MosNumID;
  NumMOS = nrow(MosNumID);

  /* open MOS SQT-ASVAB sample variance-covariance -- no correction */
  use &C_Samp_MOS;

  /* create output data for range-restriction corrected validities */
  create &Ryx_CorrectRelibRange_MOS var{MOSNUMID &TESTNAMES};
```

```
  /*looping through MOSs listed under MOSTextID*/
  do idxMOS=1 to NumMOS;

    /* read reliability corrected ASVAB-SQT covariance, uncorrected ASVAB
       variance-covariance, and uncorrected SQT variance from iTH MOS     */
    setin &Cyx_CorrectRelib_MOS;
    read all var{&TESTNAMES} where(MOSNUMID=idxMOS) into Cxc; *correctedCVxy;
    setin &C_Samp_MOS;
    read all var{&TESTNAMES} where(names?TestNames & MOSNUMID=idxMOS) into Cxx;
    read all var{SQT} where(names='SQT' & MOSNUMID=idxMOS) into Cyy;

    /* compute range-restriction corrected ASVAB-SQT covariances for iTH MOS */
    PopCxc = PopCxx*inv(Cxx)*Cxc`;
    PopCcc = Cyy+Cxc*inv(Cxx)*(PopCxc-Cxc`);

    /* compute range-restriction corrected ASVAB-SQT correlations for iTH MOS*/
    PopRxc = (1/SDx/*Sxvec*/)#(PopCxc)#(1/sqrt(PopCcc));

    /* append iTH MOS SQT-ASVAT correlations to output data */
    TmpOutput = idxMOS || PopRxc`;
    setout &Ryx_CorrectRelibRange_MOS;
    append from TmpOutput;

  end;

  close &Cyx_CorrectRelib_MOS;
  close &C_Samp_MOS;
  close &Ryx_CorrectRelibRange_MOS;

quit;
run;
```

Step 2: Aggregation of ASVAB Subtests-SQT Validities by Job Family

```
/*****************************************************************************
 Aggregating corrected ASVAB-SQT validity coefficients by job family.
*****************************************************************************/

%macro JFValid(JF_VALIDITY_DATA,MOS_VALIDITY_DATA,JF_SOLUTION,JF_CONFIG_DATA);

/* variable names for ASVAB subtests                                        */
%let TESTNAMES=GS AR NO CS AS MK MC EI VE;

proc iml;

  /* open data containing Job Family MOS configuration */
  use &JF_CONFIG_DATA;
  read all var{&JF_SOLUTION} into JFSolVec;

  /* Total number of JF in JFSolVec vector */
  NumJF = max(JFSolVec);

  /* open data containing reference population MOS validities */
  use &MOS_VALIDITY_DATA;

  /* create output data set for aggregated JF validities */
  create  &JF_VALIDITY_DATA var{&JF_SOLUTION &TESTNAMES};
  setout &JF_VALIDITY_DATA;

  do idxJF = 1 to NumJF;

    /* locate MOS in iTH job family and read acquisition weights */
    setin &JF_CONFIG_DATA;
    MOSJFIDX = loc(JFSolVec=idxJF);
    read point(MOSJFIDX) var{AcqN} into N_Wgt;

    /* read corrected validities of MOSs in iTH job family */
    setin &MOS_VALIDITY_DATA;
    read point(MOSJFIDX) var{&TESTNAMES} into XYvec;

    /* aggregate validity coefficients across MOS weighted by N          */
    /* - note job family index is concatenated to output validity vector */
    JFCorr = idxJF || (diag(N_Wgt)*XYvec)[+,]/sum(N_Wgt);

    append from JFCorr;

    end;

  close &JF_CONFIG_DATA;
  close &MOS_VALIDITY_DATA;
  close &JF_VALIDITY_DATA;

quit;
run;

%mend;
```

```
option mprint=1;

libname DirPath "D:\NewAA\";

/* 9 JF validities corrected to youth population */
%JFValid(DirPath.JF9YouthValid,DirPath.Youthvalid,JF9,DirPath.EntryMOS155);

/* 17 JF validities corrected to youth population */
%JFValid(DirPath.JF17YouthValid,DirPath.Youthvalid,JF17,DirPath.EntryMOS155);

/* 150 JF validities corrected to Army input population */
%JFValid(DirPath.JF150ArmyValid,DirPath.Armyvalid,JF150,DirPath.EntryMOS155);
```

Step 3: Calculation of Population Beta Weights by Job Family

```
/*****************************************************************************
 Computing Beta Weights by job family.
 * Use macro argument CONSTRAINT to obtain different solutions:
       NONE = no constraint on subtest weights
   POSITIVE1 = Postive weights using Zeidner-Johnson-Vladimirsky stopping rule
   POSITIVE2 = Postive weights -- ignoring solutions with negative weights
*****************************************************************************/

/* variable names for ASVAB subtests, excluding NO and CS */
%let TESTNAMES=GS AR AS MK MC EI VE;


%macro BetaWeights(BETADATA,COVDATA,VALIDITYDATA,JFSOLUTION,CONSTRAINT);

%let CORRDATA=TMPCORRDATA;

proc iml;
  TestNames = {&TESTNAMES SQT};
  NTests = ncol(TestNames);
  _TYPE_={"MEAN","STD","N"}//j(NTests,1,"CORR");
  _NAME_=j(3,1,"")//t(TestNames);

  call symput('MNTESTS',char(NTests));

  /* Used later for CONSTRAINT=POSITIVE */
  call symput('MNTESTS_ASVAB',char(NTests-1));

  do i=1 to NTests;
    if(i<10) then
        MTESTNAME = concat('MTESTNAME',char(i,1,0));
    else
        MTESTNAME = concat('MTESTNAME',char(i,2,0));
    call symput(MTESTNAME,TestNames[i]);
  end;

  use &COVDATA;
  read all var{&TESTNAMES} where((Names?TestNames) & (Names^?"SQT")) into RXX;
  close &COVDATA;

  SXX_INV = sqrt(diag(1/RXX));
  RXX = SXX_INV*RXX*SXX_INV;

  XMEAN = j(1,NTests,0);
  XSTD  = j(1,NTests,1);

  /* NOT actual sample sizes, but does not matter for estimation */
  XN    = j(1,NTests,10000);

  /* Read Validity Data Matrix -- Note that MOS/JF<->Row */
  use &VALIDITYDATA;
  read all var{&TESTNAMES} into RXY_ALL;
  read all var{&JFSOLUTION} into JFNO_ALL;
  close &VALIDITYDATA;

  /* For each job family, read validities and create correlation matrix */
  create &CORRDATA(Type=corr) var ({ &JFSOLUTION _TYPE_ _NAME_}||TestNames);
  do iJF = 1 to nrow(JFNO_ALL);

    IdxJF = JFNO_ALL[iJF];
```

```
    &JFSOLUTION = j(nrow(_TYPE_),1,IdxJF);

    RXY = RXY_ALL[iJF,];
    XCORR = (RXX//RXY)||(t(RXY)//1);
    XDATA = XMEAN//XSTD//XN//XCORR;

    %do i=1 %to &&MNTESTS;
      &&MTESTNAME&i = XDATA[,&i];
    %end;

    append;

  end; /* ENDOF: do iJF = 1 to nrow(JFNO_ALL) */

  close &CORRDATA;

  quit;
run;

%if &CONSTRAINT=NONE %then
  %let MODELOPTION=NOINT;
%else
  %let MODELOPTION=NOINT SELECTION=RSQUARE B;

proc reg data=&CORRDATA
         outest=&BETADATA (keep=&JFSOLUTION &TESTNAMES _RSQ_ _P_)
         NOPRINT;
  model SQT = &TESTNAMES / &MODELOPTION;
  by &JFSOLUTION;
  quit;
run;

/*******************************************************************
 Zeidner-Johnson-Vladimirsky Non-negative Beta Weights Approach
*******************************************************************/
%if &CONSTRAINT=POSITIVE1 %then %do;
  proc sort data=&BETADATA;
    by &JFSOLUTION descending _P_ descending _RSQ_;
  run;

  /* Oragnize all possible solutions using two data sets:              */
  /* TmpBetaPositive: solutions with non-negative weights              */
  /* TmpBetaMax: solutions with maximum R for each JF & no. subtests pair */
  data TmpBetaPositive
       TmpBetaMax
          (keep=&JFSOLUTION _RSQ_ _P_);
    array Beta {&MNTESTS_ASVAB} &TESTNAMES;
    set &BETADATA;
    by &JFSOLUTION descending _P_ descending _RSQ_;
    NegativeWgtFlag = 0;
    do i=1 to &MNTESTS_ASVAB;
      if (Beta{i}=.) then Beta{i} = 0;
      NegativeWgtFlag = NegativeWgtFlag or (Beta{i}<0);
    end;
    /* output all solutions without negative weights */
    if (^NegativeWgtFlag) then
      output TmpBetaPositive;
    /* output subset with maximum R overall for given number of subtests */
    if (First._P_ and ^First.&JFSOLUTION) then do;
      _P_ = _P_+1;
      output TmpBetaMax;
      end;
   run;
```

```
  /* output positive weighted solutions with R2 >= max R2 in the next level */
  data TmpCompare;
    keep &JFSOLUTION &TESTNAMES _RSQ_;
    merge TmpBetaPositive TmpBetaMax (rename=(_RSQ_=Rmax));
    by &JFSOLUTION descending _P_;
    if(_RSQ_ >= Rmax) then output;
  run;

  /* output only weights with maximum number of subsets for job family */
  data &BETADATA;
    set TmpCompare;
    by &JFSOLUTION descending _RSQ_;
    if First.&JFSOLUTION then output;
  run;
%end;

/*******************************************************************
 HumRRO Simple Non-negative Beta Weights Approach:
 - Entirely ignore solutions with negative weights.
*******************************************************************/
%if &CONSTRAINT=POSITIVE2 %then %do;
  /* keep only solutions with all positive weights */
  data TmpBetaPositive;
    array Beta {&MNTESTS_ASVAB} &TESTNAMES;
    set &BETADATA;
    do i=1 to &MNTESTS_ASVAB;
      if (Beta{i}=.) then Beta{i} = 0;
      else if (Beta{i}<0) then delete;
    end;
  run;

  proc sort data=TmpBetaPositive;
    by &JFSOLUTION descending _RSQ_;
  run;

  /* output all positive weights solution with maximum R2 */
  data &BETADATA;
    keep &JFSOLUTION &TESTNAMES _RSQ_;
    set TmpBetaPositive;
    by &JFSOLUTION descending _RSQ_;
    if First.&JFSOLUTION then output;
  run;
%end;

%mend;


option mprint=1;

libname DirPath "D:\NewAA\";

%BetaWeights(Modlib.JF9Beta,Modlib.PopCovYouth,Modlib.Jf9youthvalid,JF9,POSITIVE1);
%BetaWeights(Modlib.JF17Beta,Modlib.PopCovYouth,Modlib.Jf17youthvalid,JF17,POSITIVE1);
%BetaWeights(Modlib.JF150Beta,Modlib.PopCovArmy,Modlib.Jf150ArmyValid,JF150,NONE);
```

Step 4-5: Calculation of Operational ASVAB Subtest Weights and Constant

```
/*****************************************************************************
 Computing Beta Weights by job family.
*****************************************************************************/

/* variable names for ASVAB subtests, excluding NO and CS */
%let TESTNAMES=GS AR AS MK MC EI VE;


%macro UKWeights(UKDATA,COVDATA,POPDATA,BETADATA,JFNUM,TYPE);
proc iml;

  Subtest = {&TESTNAMES};

  /* predictor correlation matrix for Army Input Population*/
  use &COVDATA;
  read all var{&TESTNAMES} where(names?Subtest) into CovMat;
  close &COVDATA;

  use &POPDATA;
  read all var{SD} where(test?Subtest) into SDvec;
  %if &TYPE=TIER1 %then %do;
    read all var{MEAN} where(test?Subtest) into Means;
  %end;
  close &POPDATA;

  SDProd = 1/(SDvec#SDvec);
  R = CovMat#SDProd`;

  /*reading in all JFs into JFNO_ALL*/
  use &BETADATA;
  read all var{&JFNUM} into JFNO_ALL;
  NumJF = nrow(JFNO_ALL);

  /*creating SAS dataset containing u weights and k values for all JFs */
  create &UKDATA var{JFNO &TESTNAMES k};

  do idxJF=1 to NumJF;

    /*converting beta weights to b-weights for MOS-level*/

    setin &BETADATA;
    read all var{&TESTNAMES} where(&JFNUM=idxJF) into ObsBeta;

    bweights = ObsBeta#(1/SDvec`);

    /*transform b-weights to u and k values for Tier2 */
    %if &TYPE=TIER2 %then %do;
        /* composite multiplier*/
        CM = 20/(10*(SQRT(bweights*R*bweights`)));
        /* calculate U and K values */
        Uvec = diag(CM)*bweights;
        K = (SUM(Uvec)*50)-100;
    %end;

    /*transform b-weights to u and k values for Tier1 */
    %else %if &TYPE=TIER1 %then %do;
        /* calculate U and K values */
        Uvec = bweights;
```

```
        /* sum ASVAB means weighted by their respective b-weight*/
        K = SUM(Uvec#Means`);
    %end;

    /*merging u values with k value and adding a column for JFNO*/
    UKvec = J(nrow(K),1,idxJF) || Uvec || K;

    append from UKvec;

  end;

  close &UKDATA;
  close &BETADATA;

quit;
run;

%mend;

option mprint=1;

libname DirPath "D:\NewAA\";

/* 9 JF operational weights corrected to youth population */
%UKWeights(DirPath.JFby9uk,DirPath.PopCovYouth,DirPath.PopDescripYouth,
DirPath.JF9Beta,JF9,TIER2)

/* 17 JF operational weights corrected to youth population */
%UKWeights(DirPath.JFby17uk,DirPath.PopCovYouth,DirPath.PopDescripYouth,
DirPath.JF17Beta,JF17,TIER2)

/* 9 JF operational weights corrected to Army input population */
%UKWeights(DirPath.JFby150uk,DirPath.PopCovArmy,DirPath.PopDescripArmy,
DirPath.JF150Beta,JF150,TIER1)
```

APPENDIX B: ORIGINAL PROGRAMS IMPLEMENTING ZEIDNER, JOHNSON, AND COLLEAGUES (2000) METHOD

Part A: Step 1

```
/*******************************************************************************************
        Compute basic descriptives for 155 MOS and Army Input population;
    Correct ASVAB-SQT validities for criterion unreliability and range restriction.
*******************************************************************************************/

new;

NAME = {     /* List of 155 MOS names */
"11B" ,
"11C" ,
*...(con.)...,*
"98G" ,
"98H" ,
"98Z"
 } ;

/* Load vector of SQT reliabilities */
LOAD rlblt[] = A\data\reliabty;

/* Determine number of MOS in vector of SQT reliabilities */
JOBS = ROWS( RLBLT );

/* Set ASVAB subtest names for output */
test = { "GS     " "AR     " "NO     " "CS     " "AS     " "MK     "
         "MC     " "EI     " "VE     " "SQT    " };

/* Initialize a corrected population matrix */
Rt = zeros( 10, 10 );

/* Index vector for the explicit variables - 9 ASVAB subtests */
indx = { 1 2 3 4 5 6 7 8 9 };

/* Index for the implicit variables - SQT */
indy = { 10 };


/* Combined vector for explicit and implicit variables */
indt = indx ~ indy;

/* Remove existing Army data file in order to avoid appending the same data */
DOS DEL ARMY\ARMY.DAT;

/*****************************************************************************************
Create Army Input population data file and compute basic descriptives and ASVAB
intercorrelation matrix
*****************************************************************************************/

I = 1;
DO WHILE I <= JOBS; /* Beginning of loop */

/*  Create DOS command to append MOS file */
    CMD = "TYPE JOBS\\" $+ NAME[ I ] $+ " >> ARMY\\ARMY.DAT";
    DOS ^CMD; /* Execute the above command */

    I = I + 1;

ENDO;  /* End of loop. File ARMY.DAT contains data from all MOS */
```

```
    f1 = "ARMY\\ARMY.DAT";

    load X[ ] = ^f1;              /* Load the Army Input population data file */
    nrws = rows( X ) / 10;        /* Compute N for Army Input population */
    X = reshape( X, nrws, 10 ); /* Reshape the matrix */
    SD_a = stdc( X );             /* Compute SDs for Army Input population */
    MEAN_a = meanc( X );          /* Compute means for Army Input population */

    format/rd 21,16;

/* Output SDs */
    output file = "army\\SD" reset;
    print SD_a;

/* Output means */
    output file = "ARMY\\MEAN" reset;
    print MEAN_a;
    output off;

/* Compute variance-covariance matrix for 9 ASVAB subtests */
    VCxx = vcx( X[ ., indx ] );

    /* Output ASVAB variance-covariance matrix */
    screen off;
    format/rd 20,16;
    OUTPUT FILE = ARMY\VAR_COV.ASC RESET;
    PRINT VCxx;

/* Compute ASVAB intercorrelation matrix from variance-covariance matrix */
    Gxx = corrvc( VCxx );

    /* Output intercorrelation matrix */
    output file = "ARMY\\CORR_ARM.ASC" reset;
    print Gxx;

    CLEAR X;  /* Remove matrix of Army Input data from memory to free space */

NUMS=ZEROS( JOBS,1);  /* Initialize vector of observed MOS sample sizes (N) */


/*************************************************************************************/
/*** Process all 155 MOS. Compute corrected ASVAB-SQT validities for all MOS *********/
/*************************************************************************************/

N = 0;
l = 1;
do while l <= jobs;

    format/rd 20,16;

    f1 = "JOBS\\" $+ name[ l ];
    load X[ ] = ^f1;              /* Load matrix of ASVAB-SQT data for current MOS */
    nrws = rows( X ) / 10;        /* Find number of rows in the data matrix */
    NUMS[ L ] = NRWS;             /* Update vector of MOS sample sizes */
    X = reshape( X, nrws, 10 ); /* Reshape the data matrix */
    MEAN = MEANC( X );            /* Compute means for current MOS */
    SD = stdc( X );               /* Compute SDs for current MOS */

/* Compute ASVAB-SQT variance-covariance matrix (VCV) for current MOS */
    VC_l = vcx( X );
/* Compute correlation matrix from VCV matrix for MOS */
    RV_l = corrvc( VC_l );
/* Extract 9 x 9 submatrix corresponding to ASVAB subtests from VCV matrix */
```

```
    Gxx_1 = VC_1[ indx, indx ];
/* Extract 9 x 1 submatrix corresponding to ASVAB-SQT validities from VCV matrix */
    Gxy = VC_1[ indx, indy ];
/* Correct ASVAB-SQT validities for criterion unreliability */
    Gxy = Gxy / sqrt( rlblt[ 1 ] );

/* Correction for restriction in range to Army Input Population */
    Gyy = VC_1[ indy, indy ];  /* Extract variance for SQT */
   /* Put together VCV matrix and compute correlation matrix */
    RV = corrvc( (( VCxx~Gxy ) | ( Gxy'~Gyy )) );

   /* Use correction method described in literature */
    Gxy_a = VCxx * inv( Gxx_1 ) * Gxy;
    Gyy_a = Gyy + Gxy' * inv( Gxx_1 ) * ( Gxy_a - Gxy );
    G_a = ( VCxx ~ Gxy_a ) | ( Gxy_a' ~ Gyy_a );  /* G_a is corrected VCV matrix */
   /* Compute correlation matrix from range restricted corrected VCV matrix */
    RV_a =  corrvc(G_a);

   /*Output ASVAB-SQT validities corrected to Army Input population */
    screen off;
    FORMAT/RD 20,16;
    fl = "ARMY\\" $+ name[ 1 ] $+ ".VLD";
    output file = ^fl reset;
    print RV_a[ indy, indx ];
    OUTPUT OFF;


/* Correction for restriction in range to 1980 Youth population */
   /* Load ASVAB intercorrelation matrix for 1980 Youth from the literature */
    load VC_y[ 9, 9 ] = \YEFIM\ARI\corr.dat;
   /* Multiply correlations by variance of the Youth population, which is equal 100 */
    VC_y = VC_y * 100.;

   /* Use correction method from literature */
    Gxy_y = VC_y * inv( Gxx_1 ) * Gxy;
    Gyy_y = Gyy_a + Gxy_a' * inv( VCxx ) * ( Gxy_y - Gxy_a );
    G_y = ( VC_y ~ Gxy_y ) | ( Gxy_y' ~ Gyy_y );   /* G_y is corrected VCV matrix */
    /* Compute correlation matrix from range restricted corrected VCV matrix */
    RV_y = corrvc( G_y );

    /* Output ASVAB-SQT validities corrected to 1980 Youth Population */
    screen off;
    FORMAT/RD 20,16;
    fl = "YOUTH\\" $+ name[ 1 ] $+ ".VLD";
    output file = ^fl reset;
    print RV_y[ indy, indx ];
    OUTPUT OFF;

/* Output of ASVAB-SQT validities for validation check */
    fl = "JOBS\\" $+ name[ 1 ] $+ ".CHK";
    SCREEN OFF;
    output file = ^fl reset;
    FORMAT 3,0;
    print;
    PRINT $name[ 1 ] ;
    print "  Uncrr   Atten   Army   Youth    STD   MEAN";
    print;
    FORMAT/RD 7,4;
    K = 1;
    DO WHILE K <= 10;
    print RV_1[K,10]~RV[K,10]~RV_a[K,10]~RV_y[K,10]~SD[K]~MEAN[K] $TEST[K];
    K = K + 1;
    ENDO;
```

```
/* Check for non-positivity, i.e., correlation matrices must be non-singular. */
    d = det( RV_a );
    if d > 0;
      ;
    else;
    screen on;
    print $name[ l ];
    print "Determinant  " d;
    screen off;
    endif;

    l = l + 1;

endo;

/************************  End of processing of all MOS  ******************************/

output off;
screen on;
end;
```

```
NEW;

REPS = 0;

SAMPL = 1000;

/* Set flag for negativity or positivity of composite weights */
NEGATIVE = 0 ; @ 1 - with negatives, 0 - without negatives @

/* Initialize sets of indices for different ASVAB batteries */
ALLtests = { 1,2,3,4,5,6,7,8,9};            @ All tests @
NO_out = { 1, 2, 4, 5, 6, 7, 8, 9 };      @ Without NO test @
CS_out = { 1, 2, 3, 5, 6, 7, 8, 9 };      @ Without CS test @
NOCS_out = { 1, 2, 5, 6, 7, 8, 9 };       @ Without NO and CS tests @

TESTS = nocs_out;          /* This one excludes NO and CS */

N_tests = ROWS( TESTS );  /* Determine number of ASVAB subtests */

/* Define full names for all 155 MOS */
MOS = {
"11B" "Infantry" "man      " "          " "         " "         "          " ,
"11C" "Indirect" " Fire In" "fantryma" "n         " "         "          " ,
*...(con.)...*
"98G" "EW Signa" "l Intell" "igence V" "oice Int" "errogato" "r          " ,
"98H" "Morse In" "tercepto" "r        " "          " "         "          " ,
"98Z" "Emitter " "Locator/" "Identifi" "er         " "         "          "
 } ;

indx = { 1 2 3 4 5 6 7 8 9 };    /* Index for the explicit variables */

indy = { 10 };                   /* Index for the implicit variables */

indt = indx ~ indy;


/* Load precomputed cluster solution */
LOAD CLUST[] = CLUST.OUT;

/* Determine number of MOS */
JOBS = ROWS( CLUST );

/* Load acquisition numbers based on Seabrook Report (1989) */
LOAD NUM[] = \YEFIM\ARI\1996\a\DATA\acquisit;

/* Determine number of job families by the largest cluster number */
FMLS = MAXC( CLUST );

/* Create character names for clusters, i.e., 1, 2, ... */
NAME = SEQA( 1, 1, FMLS );
NAME = 0 $+ FTOCV( NAME , 3,0 );

/* Start of Step 2*/

  FORMAT/RD 20, 16;

/* Set output file for aggregated ASVAB-SQT validities */
  OUTPUT FILE = FMLS\SQT RESET;
/* Initialize quota for job families */
```

```
  QUOTA = ZEROS( FMLS, 1 );

/* Aggregate ASVAB-SQT validities to JF-level using cluster solution */
  L = 1;
  DO WHILE L <= FMLS;

  V_f = zeros( 9,1 );

  N = 0;
  K = 1;
  DO WHILE K <= JOBS;

    IF CLUST[ K ] == L;
      /* If Youth is used to create visible system */
      fl = "YOUTH\\" $+ mos[ K, 1 ] $+ ".VLD";
      /* If Army is used to create invisible system */
    /*    fl = "ARMY\\" $+ mos[ K, 1 ] $+ ".VLD";
      */
        load RV_p[ ] = ^fl;

        V_f = V_f + RV_p * NUM[ K ];

        N = N + NUM[ K ];

    ENDIF;

    K = K + 1;

  endo;

  /* Output aggregated ASVAB-SQT validities */
  SCREEN OFF;
  if N > 0;
    print V_f/N;
  else;
    print V_f/5000;
  endif;

  QUOTA[ L ] = N;  /* Update job family quota */

  L = L + 1;

  ENDO;

/*********** End of aggregating MOS into families   ********************/

/* End of Step 2 */

  output off;

/*************************************************************************************
                            Computation of Composites
**************************************************************************************/

/* Start of Step 3 */

test = { "  GS    " "  AR    " "  NO    " "  CS    " "  AS    "
         "  MK    " "  MC    " "  EI    " "  VE    " };

test = test[ TESTS];

nmb = {  "   1  " "   2  " "   3  " "   4  " "   5  "
         "   6  " "   7  " "   8  " "   9  " "      " };
```

```
y = date;
format/ldn 1,0;
output file = "OUT.CRT" reset;
SCREEN ON;
PRINT "                                                              "
                                                   y[3] "/" y[2] "/" y[1];
print
"           TEST COMPOSIT FOR SQT . ( SAMPLE  A+B+C  ).";

format/rdn 8,3;
PRINT;
print "      " $test;

test_id = { 2 3 4 5 6 7 8 9 };

load V[ ] = "FMLS\\SQT";
FMLS = ROWS( V ) / 9;
V = reshape( V, FMLS, 9 );

/* This file is used for invisible tier */
/*load RV_FULL[ 9, 9 ] = "ARMY\\CORR_ARM.ASC";*/
/* This file is used for visible tier */
load RV_FULL[ 9, 9 ] = "\\YEFIM\\ARI\\CORR.DAT";

/* Extract ASVAB intercorrelation matrix from full 9 x 9 matrix  */
RV = RV_full[ TESTS, TESTS];
/* Initialize matrix of weights (us) */
W = zeros( N_tests, FMLS );
/* Initialize matrix of Us */
U = W;
/* Initialize matrix of Ks */
KU = ZEROS( 1, FMLS );

load MEAN[ 9, 1 ] = "ARMY\\MEAN";  /* Load vector of means for all 9 ASVAB subtests */
MEAN = MEAN[ TESTS, .];            /* Extract means for specified ASVAB battery */

load SD[ 9, 1 ] = "ARMY\\SD";      /* Same as for the means */
SD = SD[ TESTS, .];                /* Same as for the means */

/* Compute diagonal matrix of reciprocal to SDs */
SD = DIAGRV( zeros( N_tests, N_tests ), 1/SD );

S = DIAGRV( ZEROS( FMLS, FMLS ), 1/SQRT( DIAG(V[.,tests]*INV(RV)*V[.,tests]')));

/* Estimate composite weights for all families */
l = 1;
do while l <= FMLS;  /* Start of loop */

    V_l = V[ L, tests ];      /* Extract SQT validities for given set of ASVAB subtests */
    W_CT = inv( RV ) * V_l'; /* Estimate Beta weights for given set of ASVAB subtests */

    PV_CTP = sqrt( V_L * W_CT) ;
    MV = PV_CTP;              /* Compute multiple R */

/* Find composite without negative weights */
        if negative == 0;
          ko = 1;
          do while ko <= N_tests;
            if W_CT[ ko ] < 0;
              RV_0 = zeros( 9, 9 );
              RV_0[ TESTS, TESTS ] = RV;
              { W_CT, MV } = select_8( RV_0, V[ L, . ] );
```

```
                W_CT = W_CT[ TESTS ];
              break;
            endif;
            ko = ko + 1;
          endo;
        endif;

    W[ ., 1 ] = W_CT;          /* Update matrix of weights */

  /* Determine order of weights in composite */
    format/rd 6,3;
    PV_CT = ones( 1, N_tests ) *10;
    k = 1;
    do while k <= N_tests;
        order = 1;
        kk = 1;
        do while kk <= N_tests;
            if W[ k, 1 ] < W[ kk, 1 ];
                order = order + 1;
            endif;
            kk = kk + 1;
        endo;

        PV_CT[ k ] = order;

        IF W[ K, L ] == 0. ;
            PV_CT[ K ] = 10;
        ENDIF;
        k = k + 1;
    endo;
```

/* End of Step 3 */

/* Start of Steps 4 & 5 */

```
/* Transform Beta weights to Us and Ks for computing AA composites scores, with mean 0
and SD 1, for invisible tier */
    U[ ., L ] = SD * W[ ., L ];
    KU[ L ] = MEAN' * U[ ., L ];

    output file = "OUT.CRT" on;

    SCREEN ON;

    FORMAT/RDN 5,0;
    print 1 ;
    FORMAT/RDN 8,5;
    PRINT " " MV;
    PRINT "      " $NMB[ PV_CT ];
    print "      " W[ ., 1 ]';
    1 = 1 + 1;
    OUTPUT OFF;
    endo;

/*  Transform Beta weights to Us and Ks for computing AA standard scores, with mean 100
and SD 20, for visible tier */
    U = W * S * 2.;
    KU = 50 * ONES( 1, N_tests ) * U - 100.;

    OUTPUT FILE = "U.OUT" RESET;
    PRINT U';
    OUTPUT OFF;
    OUTPUT FILE = "K.OUT" RESET;
```

```
    print KU';
    OUTPUT OFF;

closeall;
screen off;
```

/* Step 3 (con.) */

```
/* Below are the procedures used to eliminate negative weights from visible tier
composites */


proc ( 2 ) = select_6( RV, V );

local W_CTP, W_CT, PV_CTP, PV_CT, id, flag, n,m,i,j,k,b,c,num, e,test_id;

test_id = { 2 3 4 5 6 7 };
    num = 1;

    PV_CTP = zeros( 130, 9 );
    PV_CT = zeros( 1, 9 );
    W_CTP = zeros( 9, 1 );
    id = zeros( 1, 6 );

    c = 1;
    do while c <= 4;
      n = c + 1;
      do while n <= 5;
        m = n + 1;
        do while m <= 6;
          i = m + 1;
          do while i <= 7;
           j = i + 1;
           do while j <= 8;
             k = j + 1;
             do while k <= 9;


          id[ 1 ] = c;
          id[ 2 ] = n;
          id[ 3 ] = m;
          id[ 4 ] = i;
          id[ 5 ] = j;
          id[ 6 ] = k;

          if det( RV[ id, id ] ) == 0;
            k = k + 1;
            continue;
          endif;

          W_CT = inv( RV[ id, id ] ) * V[ 1, id ]';

          PV_CTP[ num, 1 ] = sqrt( V[ 1, id ] * W_CT) ;

           PV_CTP[ num, test_id ] = id;

           if num == 1;
              num = num + 1;
              continue;
           endif;

           if PV_CT[ 1 ] < PV_CTP[ num, 1 ];
```

```
                flag = 1;
                e = 1;
                do while e <= 6;
                    if W_CT[ e ] < 0;
                        flag = -1;
                        break;
                    endif;
                    e = e + 1;
                endo;
                if flag == 1;
                    PV_CT = PV_CTP[ num, . ];
                    W_CTP = zeros( 9, 1 );
                    W_CTP[ PV_CT[ test_id ] ] = W_CT;
                endif;

           endif;

              num = num + 1;

              k = k + 1;
              endo;
            j = j + 1;
            endo;
          i = i + 1;
          endo;
        m = m + 1;
        endo;
      n = n + 1;
      endo;
    c = c + 1;
    endo;

if W_CTP*W_CTP' == 0;
    { W_CTP, PV_CT[ 1 ] } = select_5( RV, V );
endif;

retp( W_CTP, PV_CT[ 1 ] );

endp;


proc ( 2 ) = select_5( RV, V );

local W_CTP, W_CT, PV_CTP, PV_CT, id, flag, n,m,i,j,k,b,c,num, e,test_id;

test_id = { 2 3 4 5 6 };
    num = 1;

    PV_CTP = zeros( 130, 9 );
    PV_CT = zeros( 1, 9 );
    W_CTP = zeros( 9, 1 );
    id = zeros( 1, 5 );

      n = 1;
      do while n <= 5;
        m = n + 1;
        do while m <= 6;
          i = m + 1;
          do while i <= 7;
           j = i + 1;
           do while j <= 8;
             k = j + 1;
             do while k <= 9;
```

```
          id[ 1 ] = n;
          id[ 2 ] = m;
          id[ 3 ] = i;
          id[ 4 ] = j;
          id[ 5 ] = k;

          if det( RV[ id, id ] ) == 0;
            k = k + 1;
            continue;
          endif;

          W_CT = inv( RV[ id, id ] ) * V[ 1, id ]';

          PV_CTP[ num, 1 ] = sqrt( V[ 1, id ] * W_CT) ;

           PV_CTP[ num, test_id ] = id;

           if num == 1;
              num = num + 1;
              continue;
           endif;

           if PV_CT[ 1 ] < PV_CTP[ num, 1 ];

                flag = 1;
                e = 1;
                do while e <= 5;
                     if W_CT[ e ] < 0;
                          flag = -1;
                          break;
                     endif;
                     e = e + 1;
                endo;
                if flag == 1;
                     PV_CT = PV_CTP[ num, . ];
                     W_CTP = zeros( 9, 1 );
                     W_CTP[ PV_CT[ test_id ] ] = W_CT;
                endif;

           endif;

              num = num + 1;

              k = k + 1;
              endo;
            j = j + 1;
            endo;
          i = i + 1;
          endo;
        m = m + 1;
        endo;
      n = n + 1;
      endo;

if W_CTP*W_CTP' == 0;
    { W_CTP, PV_CT[ 1 ] } = select_4( RV, V );
endif;

retp( W_CTP, PV_CT[ 1 ] );

endp;
```

```
proc ( 2 ) = select_4( RV, V );

local W_CTP, W_CT, PV_CTP, PV_CT, id, flag, n,m,i,j,k,b,c,num, e,test_id;

test_id = { 2 3 4 5 };
    num = 1;

    PV_CTP = zeros( 130, 9 );
    PV_CT = zeros( 1, 9 );
    W_CTP = zeros( 9, 1 );
    id = zeros( 1, 4 );

        m =  1;
        do while m <= 6;
          i = m + 1;
          do while i <= 7;
           j = i + 1;
           do while j <= 8;
             k = j + 1;
             do while k <= 9;

          id[ 1 ] = m;
          id[ 2 ] = i;
          id[ 3 ] = j;
          id[ 4 ] = k;

          if det( RV[ id, id ] ) == 0;
            k = k + 1;
            continue;
          endif;

          W_CT = inv( RV[ id, id ] ) * V[ 1, id ]';

          PV_CTP[ num, 1 ] = sqrt( V[ 1, id ] * W_CT) ;

           PV_CTP[ num, test_id ] = id;

           if num == 1;
              num = num + 1;
              continue;
           endif;

           if PV_CT[ 1 ] < PV_CTP[ num, 1 ];

                flag = 1;
                e = 1;
                do while e <= 4;
                    if W_CT[ e ] < 0;
                        flag = -1;
                        break;
                    endif;
                    e = e + 1;
                endo;
                if flag == 1;
                    PV_CT = PV_CTP[ num, . ];
                    W_CTP = zeros( 9, 1 );
                    W_CTP[ PV_CT[ test_id ] ] = W_CT;
                endif;

           endif;

              num = num + 1;
```

```
              k = k + 1;
              endo;
            j = j + 1;
            endo;
          i = i + 1;
          endo;
        m = m + 1;
        endo;

if W_CTP*W_CTP' == 0;
    { W_CTP, PV_CT[ 1 ] } = select_3( RV, V );
endif;

retp( W_CTP, PV_CT[ 1 ] );

endp;

proc ( 2 ) = select_3( RV, V );

local W_CTP, W_CT, PV_CTP, PV_CT, id, flag, n,m,i,j,k,b,c,num, e,test_id;

test_id = { 2 3 4 };
    num = 1;

    PV_CTP = zeros( 130, 9 );
    PV_CT = zeros( 1, 9 );
    W_CTP = zeros( 9, 1 );
    id = zeros( 1, 3 );

          i = 1;
          do while i <= 7;
           j = i + 1;
           do while j <= 8;
             k = j + 1;
             do while k <= 9;

          id[ 1 ] = i;
          id[ 2 ] = j;
          id[ 3 ] = k;

          if det( RV[ id, id ] ) == 0;
            k = k + 1;
            continue;
          endif;

          W_CT = inv( RV[ id, id ] ) * V[ 1, id ]';

          PV_CTP[ num, 1 ] = sqrt( V[ 1, id ] * W_CT) ;

           PV_CTP[ num, test_id ] = id;

           if num == 1;
              num = num + 1;
              continue;
           endif;

           if PV_CT[ 1 ] < PV_CTP[ num, 1 ];

                flag = 1;
                e = 1;
                do while e <= 3;
                    if W_CT[ e ] < 0;
```

```
                flag = -1;
                break;
            endif;
            e = e + 1;
        endo;
        if flag == 1;
            PV_CT = PV_CTP[ num, . ];
            W_CTP = zeros( 9, 1 );
            W_CTP[ PV_CT[ test_id ] ] = W_CT;
        endif;

     endif;

        num = num + 1;

        k = k + 1;
        endo;
      j = j + 1;
      endo;
    i = i + 1;
    endo;

if W_CTP*W_CTP' == 0;
    { W_CTP, PV_CT[ 1 ] } = select_3( RV, V );
endif;

retp( W_CTP, PV_CT[ 1 ] );

endp;

proc ( 2 ) = select_2( RV, V );

local W_CTP, W_CT, PV_CTP, PV_CT, id, flag, n,m,i,j,k,b,c,num, e,test_id;

test_id = { 2 3 };
    num = 1;

    PV_CTP = zeros( 130, 9 );
    PV_CT = zeros( 1, 9 );
    W_CTP = zeros( 9, 1 );
    id = zeros( 1, 2 );

           j = 1;
           do while j <= 8;
             k = j + 1;
             do while k <= 9;


          id[ 1 ] = j;
          id[ 2 ] = k;

          if det( RV[ id, id ] ) == 0;
            k = k + 1;
            continue;
          endif;

          W_CT = inv( RV[ id, id ] ) * V[ 1, id ]';

          PV_CTP[ num, 1 ] = sqrt( V[ 1, id ] * W_CT) ;

           PV_CTP[ num, test_id ] = id;

           if num == 1;
```

```
            num = num + 1;
            continue;
         endif;

         if PV_CT[ 1 ] < PV_CTP[ num, 1 ];

              flag = 1;
              e = 1;
              do while e <= 2;
                  if W_CT[ e ] < 0;
                      flag = -1;
                      break;
                  endif;
                  e = e + 1;
              endo;
              if flag == 1;
                  PV_CT = PV_CTP[ num, . ];
                  W_CTP = zeros( 9, 1 );
                  W_CTP[ PV_CT[ test_id ] ] = W_CT;
              endif;

         endif;

            num = num + 1;

            k = k + 1;
            endo;
          j = j + 1;
          endo;

retp( W_CTP, PV_CT[ 1 ] );

endp;
```

/* End of Step 3 */

APPENDIX C: BASIC DESCRIPTIVES AND ASVAB SUBTEST VARIANCE-COVARIANCE (VCV) MATRICES FOR ARMY INPUT AND YOUTH POPULATIONS

Table 1
Basic Descriptives for ASVAB Subtests and SQT for Army Input Population (N = 257,810)

Variable	$M(\mu)$	$SD(\sigma)$	$VAR(\sigma^2)$
GS	52.13830340	7.55278124	57.04450447
AR	52.41040689	7.07276512	50.02400645
NO	54.06027307	6.47601979	41.93883230
CS	52.97999690	6.94181519	48.18879808
AS	53.22434351	8.83256904	78.01427576
MK	51.56189830	7.76769675	60.33711285
MC	53.75594042	8.43933343	71.22234882
EI	52.20712540	8.30057262	68.89950588
VE	52.87955471	5.58611374	31.20466676
SQT	0.00007664	1.00017246	1.00034494

Note. For Youth Population, all ASVAB subtests have mean (μ) of 50 and SD (σ) of 10; SQT mean (μ) and SD (σ) same as Army Input Population.

GS = General Science; AR = Arithmetic Reasoning; NO = Numerical Operations; CS = Coding Speed; AS = Auto & Shop Information; MK = Mathematical Knowledge; EI = Electronics Information; VE = Verbal

Table 2
ASVAB Subtest Variance-Covariance (VCV) Matrix for Army Input Population (N = 257,810)

	GS	AR	NO	CS	AS	MK	MC	EI	VE
GS	57.04450447	24.03543527	-1.99574141	0.55323992	30.08100496	27.48286504	35.46816127	37.89577453	29.83954632
AR	24.03543527	50.02400645	10.10300140	8.85037942	19.06172355	36.48989068	30.82144616	23.35250004	17.25495601
NO	-1.99574141	10.10300140	41.93883230	24.00625978	-9.45136157	14.31275736	-2.16063659	-5.03994244	-0.53673611
CS	0.55323992	8.85037942	24.00625978	48.18879808	-6.03941701	11.36504819	1.51034830	-1.39594372	3.59526961
AS	30.08100496	19.06172355	-9.45136157	-6.03941701	78.01427576	9.97406579	44.36535155	44.80651289	17.90743269
MK	27.48286504	36.48989068	14.31275736	11.36504819	9.97406579	60.33711285	28.51679811	22.49318756	18.22467046
MC	35.46816127	30.82144616	-2.16063659	1.51034830	44.36535155	28.51679811	71.22234882	41.75330783	22.60526383
EI	37.89577453	23.35250004	-5.03994244	-1.39594372	44.80651289	22.49318756	41.75330783	68.89950588	23.56569266
VE	29.83954632	17.25495601	-0.53673611	3.59526961	17.90743269	18.22467046	22.60526383	23.56569266	31.20466676

Table 3
ASVAB Subtest Variance-Covariance (VCV) Matrix for 1980 Youth Population

Variable	GS	AR	NO	CS	AS	MK	MC	EI	VE
GS	100.00	72.00	52.00	45.00	64.00	69.00	70.00	76.00	80.00
AR	72.00	100.00	63.00	51.00	53.00	83.00	69.00	66.00	73.00
NO	52.00	63.00	100.00	70.00	30.00	62.00	40.00	41.00	62.00
CS	45.00	51.00	70.00	100.00	22.00	52.00	34.00	34.00	57.00
AS	64.00	53.00	30.00	22.00	100.00	41.00	74.00	75.00	52.00
MK	69.00	83.00	62.00	52.00	41.00	100.00	60.00	59.00	70.00
MC	70.00	69.00	40.00	34.00	74.00	60.00	100.00	74.00	60.00
EI	76.00	66.00	41.00	34.00	75.00	59.00	74.00	100.00	67.00
VE	80.00	73.00	62.00	57.00	52.00	70.00	60.00	67.00	100.00

APPENDIX D: CRITERION RELIABILITIES BY MOS

Criterion Reliabilities (r_{yy}) by MOS

MOS	r_{yy}	MOS	r_{yy}	MOS	r_{yy}	MOS	r_{yy}
11B	0.860	31V	0.800	63S	0.840	88H	0.750
11C	0.880	35E	0.830	63T	0.850	88M	0.870
11H	0.780	35H	0.830	63W	0.840	88N	0.840
11M	0.850	35J	0.830	63Y	0.840	91A	0.810
12B	0.850	35N	0.830	67N	0.840	91D	0.805
12C	0.820	36M	0.830	67R	0.840	91E	0.805
12F	0.850	41C	0.830	67T	0.840	91F	0.805
13B	0.830	44B	0.860	67U	0.840	91G	0.805
13C	0.845	44E	0.850	67V	0.910	91K	0.805
13E	0.845	45B	0.830	67Y	0.840	91M	0.805
13F	0.860	45D	0.830	68B	0.850	91P	0.805
13M	0.900	45E	0.840	68D	0.840	91Q	0.805
13N	0.770	45K	0.850	68F	0.840	91R	0.805
13R	0.850	45L	0.830	68G	0.830	91S	0.805
14D	0.770	45N	0.840	68J	0.880	91T	0.805
15E	0.805	45T	0.830	68M	0.830	91Z	0.805
16E	0.805	46Z	0.850	68N	0.800	92A	0.810
16J	0.805	51B	0.850	68Z	0.830	92G	0.820
16P	0.770	51K	0.830	71D	0.870	92M	0.830
16R	0.770	51M	0.830	71G	0.840	92R	0.830
16S	0.770	51R	0.830	71L	0.850	92Y	0.890
19D	0.850	51T	0.805	71M	0.870	93C	0.805
19E	0.870	52C	0.830	72E	0.860	93F	0.830
19K	0.870	52D	0.850	72G	0.850	93P	0.805
24Z	0.830	54B	0.790	73C	0.800	95B	0.800
25M	0.720	55B	0.880	73D	0.840	95C	0.805
25S	0.820	55D	0.830	74B	0.770	96B	0.805
25Z	0.805	55G	0.830	75B	0.800	96D	0.805
27E	0.830	57E	0.830	75C	0.840	96R	0.805
27Z	0.830	62B	0.900	75D	0.780	97B	0.830
29V	0.850	62E	0.790	75E	0.840	97E	0.805
29Z	0.830	62F	0.830	75F	0.840	98C	0.805
31C	0.850	62J	0.790	76J	0.840	98G	0.805
31K	0.790	63B	0.830	76P	0.840	98H	0.805
31L	0.880	63D	0.840	76V	0.840	98Z	0.805
31N	0.830	63E	0.740	76X	0.840		
31P	0.830	63G	0.850	77F	0.850		
31Q	0.830	63H	0.840	77W	0.830		
31R	0.830	63J	0.840	81L	0.805		
31S	0.830	63N	0.840	82C	0.805		

APPENDIX E: ACQUISITION AND OBSERVED *ns* BY MOS

Acquisition and Observed *ns* by MOS

MOS	Acquisition *n*	Observed *n*	MOS	Acquisition *n*	Observed *n*
11B	13310	5000	45D	123	565
11C	1993	5000	45E	225	546
11H	2118	5000	45K	199	817
11M	338	4593	45L	105	448
12B	3560	5000	45N	45	563
12C	436	1950	45T	249	509
12F	234	603	46Z	135	498
13B	4442	5000	51B	297	2037
13C	141	720	51K	61	532
13E	632	1919	51M	28	327
13F	1274	4101	51R	77	723
13M	657	776	51T	26	344
13N	328	2724	52C	322	529
13R	166	592	52D	1233	5000
14D	30	683	54B	1152	1380
15E	408	224	55B	629	2457
16E	356	703	55D	81	415
16J	118	171	55G	110	215
16P	906	1104	57E	122	791
16R	1157	1996	62B	460	3054
16S	1880	2406	62E	504	1522
19D	5536	5000	62F	200	527
19E	232	4764	62J	406	887
19K	31	5000	63B	3250	5000
24Z	192	752	63D	359	1234
25M	23	451	63E	562	1376
25S	16	358	63G	96	785
25Z	61	372	63H	547	2396
27E	282	898	63J	383	1302
27Z	266	548	63N	116	750
29V	47	852	63S	1056	2506
29Z	222	433	63T	868	3378
31C	1723	5000	63W	1105	3062
31K	1744	5000	63Y	364	987
31L	914	2778	67N	374	1359
31N	267	709	67R	286	236
31P	173	563	67T	415	1564
31Q	423	1394	67U	268	1632
31R	1810	5000	67V	178	1751
31S	160	498	67Y	303	1168
31V	1001	4278	68B	65	640
35E	192	1021	68D	84	740
35H	75	307	68F	81	712
35J	161	1034	68G	148	904
35N	218	737	68J	258	1128
36M	223	1201	68M	90	388
41C	48	323	68N	46	475
44B	305	1045	68Z	18	749
44E	73	592	71D	240	1431
45B	93	612	71G	181	1145

Acquisition and Observed *ns* by MOS (con).

MOS	Acquisition *n*	Observed *n*	MOS	Acquisition *n*	Observed *n*
71L	1325	5000	91K	209	1478
71M	214	972	91M	93	513
72E	799	1651	91P	76	695
72G	572	1738	91Q	83	682
73C	275	2246	91R	113	558
73D	94	500	91S	119	514
74B	105	1184	91T	76	345
75B	796	4113	91Z	169	641
75C	148	2505	92A	2336	5000
75D	115	2714	92G	3002	5000
75E	211	1379	92M	33	298
75F	163	624	92R	296	1009
76J	71	997	92Y	2031	5000
76P	671	2897	93C	278	626
76V	1267	5000	93F	90	303
76X	228	541	93P	220	1327
77F	1326	5000	95B	4187	5000
77W	71	805	95C	87	323
81L	42	331	96B	486	818
82C	461	808	96D	180	361
88H	409	1525	96R	257	792
88M	4504	5000	97B	347	429
88N	99	1954	97E	323	372
91A	4601	5000	98C	840	562
91D	185	748	98G	1309	1242
91E	223	1209	98H	177	966
91F	57	474	98Z	461	463
91G	73	309			

Note. Acquisition numbers (N) based on 1989 Seabrook Reports. Observed N computed from SQT database (N = 257,810) used by Zeidner, Johnson, and colleagues (2000), also used in current replication.

APPENDIX F: RESULTS OF REPLICATION

Table 1
Population Beta Weights Corrected to the 1980 Youth Population by Army Aptitude (AA) Composite (9 Job Families)

Composite	GS	AR	AS	MK	MC	EI	VE
CL	0.00000000	0.24540772	0.02709166	0.18816961	0.03626410	0.02708013	0.24054988
CO	0.05217442	0.08871463	0.12230491	0.16687189	0.09932356	0.05728945	0.08820356
EL	0.02613946	0.14188628	0.13071219	0.15443311	0.08129539	0.10375059	0.17345885
FA	0.04164027	0.11954300	0.11243136	0.16709168	0.11698648	0.04957620	0.09799372
GM	0.07546379	0.15184787	0.18348277	0.14563193	0.09232459	0.10585534	0.07644407
MM	0.02018868	0.11387138	0.33563979	0.09711230	0.13224675	0.11424816	0.07952995
OF	0.04323428	0.16562006	0.17212560	0.10327540	0.10942997	0.06495191	0.12295242
SC	0.00377482	0.13354737	0.08517063	0.19492673	0.07528012	0.10740604	0.17844892
ST	0.04078145	0.15861362	0.07796219	0.15195751	0.09722104	0.05016700	0.21814622

Note. GS = General Science; AR = Arithmetic Reasoning; NO = Numerical Operations; CS = Coding Speed; AS = Auto & Shop Information; MK = Mathematical Knowledge; EI = Electronics Information; VE = Verbal.

CL = Clerical; CO = Combat; EL = Electronics Repair; FA = Field Artillery; GM = General Maintenance; MM = Mechanical Maintenance; OF = Operators/Food; SC = Surveillance and Communications; ST = Skilled Technical.

Table 2
Operational Weights and Constants for Computing Army Aptitude (AA) Standard Scores by AA Composite (9 Job Families)

Composite	GS	AR	AS	MK	MC	EI	VE	Constant
CL	0.00000000	0.72448885	0.07997958	0.55551137	0.10705830	0.07994555	0.71014761	12.85656224
CO	0.18428367	0.31334625	0.43198942	0.58940312	0.35081772	0.20235033	0.31154110	19.18658098
EL	0.07647172	0.41509224	0.38240212	0.45179833	0.23783193	0.30352521	0.50745868	18.72901174
FA	0.14018903	0.40246179	0.37851925	0.56254251	0.39385484	0.16690670	0.32991249	18.71933070
GM	0.21608285	0.43480084	0.52538413	0.41700213	0.26436202	0.30310592	0.21888978	18.98138275
MM	0.05343197	0.30137537	0.88831422	0.25702031	0.35000817	0.30237258	0.21048632	18.15044685
OF	0.13160275	0.50413824	0.52394073	0.31436456	0.33309875	0.19771000	0.37426031	18.95576672
SC	0.01142475	0.40419005	0.25777461	0.58995881	0.22784032	0.32507156	0.54008761	17.81738514
ST	0.12016020	0.46734586	0.22971108	0.44773401	0.28645619	0.14781417	0.64275524	17.09883705

Table 3
Population Beta Weights Corrected to the 1980 Youth Population by Army Aptitude (AA) Composite (17 Job Families)

Composite	GS	AR	AS	MK	MC	EI	VE
CL1	0.00000000	0.24818270	0.00000000	0.22718282	0.00000000	0.00000000	0.31784687
CL2	0.00000000	0.23471688	0.06118002	0.16707584	0.06309115	0.04109346	0.18704274
CO1	0.04622850	0.07153084	0.11741642	0.17833439	0.09494484	0.04672735	0.07103920
CO2	0.06270323	0.11914310	0.13096127	0.14657450	0.10707724	0.07599245	0.11859760
EL1	0.00921053	0.13745876	0.16330157	0.13198248	0.08274267	0.11752091	0.16277002
EL2	0.04468357	0.12111499	0.12948177	0.19040515	0.09021013	0.10586813	0.12851111
EL3	0.02906170	0.19212752	0.05852517	0.13633379	0.06076890	0.06815552	0.28471265
FA	0.04164027	0.11954300	0.11243136	0.16709168	0.11698648	0.04957620	0.09799372
GM1	0.06902522	0.20618259	0.19221595	0.12653791	0.08838131	0.14099515	0.08545673
GM2	0.08110807	0.10421622	0.17582696	0.16237039	0.09578140	0.07505059	0.06854328
MM1	0.02396093	0.10285152	0.36850881	0.08824955	0.13230042	0.12667491	0.05043867
MM2	0.00402387	0.16109364	0.19478959	0.13509092	0.13201677	0.06099710	0.20419174
OF	0.04323428	0.16562006	0.17212560	0.10327540	0.10942997	0.06495191	0.12295242
SC	0.00377482	0.13354737	0.08517063	0.19492673	0.07528012	0.10740604	0.17844892
ST1	0.06230523	0.14140888	0.07283551	0.12003187	0.10512087	0.05183379	0.15959357
ST2	0.03363149	0.18822437	0.04939113	0.16886138	0.07484071	0.03299229	0.27959832
ST3	0.02568328	0.15372511	0.10304183	0.16988257	0.10569916	0.06079261	0.22941320

Note. *GS = General Science; AR = Arithmetic Reasoning; NO = Numerical Operations; CS = Coding Speed; AS = Auto & Shop Information; MK = Mathematical Knowledge; EI = Electronics Information; VE = Verbal.*

CL = Clerical; CO = Combat; EL = Electronics Repair; FA = Field Artillery; GM = General Maintenance; MM = Mechanical Maintenance; OF = Operators/Food; SC = Surveillance and Communications; ST = Skilled Technical.

Table 4
Operational Weights and Constants for Computing Army Aptitude (AA) Standard Scores by AA Composite (17 Job Families)

Composite	GS	AR	AS	MK	MC	EI	VE	Constant
CL1	0.00000000	0.68564204	0.00000000	0.62762672	0.00000000	0.00000000	0.87809980	9.56842796
CL2	0.00000000	0.71792549	0.18713053	0.51103272	0.19297609	0.12569205	0.57210523	15.34310529
CO1	0.17608191	0.27245714	0.44723288	0.67926619	0.36163984	0.17798198	0.27058454	19.26222366
CO2	0.19596598	0.37235711	0.40929235	0.45808829	0.33464778	0.23749870	0.37065226	18.92512342
EL1	0.02727560	0.40706335	0.48359292	0.39084617	0.24502992	0.34802042	0.48201882	19.19236006
EL2	0.13113305	0.35543663	0.37999067	0.55878274	0.26474004	0.31069162	0.37714207	18.89584087
EL3	0.08095384	0.53518749	0.16302682	0.37976931	0.16927692	0.18985298	0.79309121	15.55792840
FA	0.14018903	0.40246179	0.37851925	0.56254251	0.39385484	0.16690670	0.32991249	18.71933070
GM1	0.18000417	0.53768357	0.50126134	0.32998595	0.23048106	0.36768757	0.22285431	18.49789842
GM2	0.25377717	0.32607975	0.55014094	0.50803699	0.29968821	0.23482407	0.21446348	19.35053085
MM1	0.06285076	0.26978487	0.96661776	0.23148317	0.34703087	0.33227488	0.13230326	17.11727902
MM2	0.01076023	0.43078076	0.52088716	0.36124684	0.35302625	0.16311244	0.54602945	19.29215697
OF	0.13160275	0.50413824	0.52394073	0.31436456	0.33309875	0.19771000	0.37426031	18.95576672
SC	0.01142475	0.40419005	0.25777461	0.58995881	0.22784032	0.32507156	0.54008761	17.81738514
ST1	0.20522752	0.46578740	0.23991323	0.39537357	0.34625813	0.17073557	0.52568604	17.44907288
ST2	0.09355802	0.52361339	0.13739911	0.46974832	0.20819620	0.09177985	0.77780273	15.10488055
ST3	0.07134240	0.42701389	0.28622710	0.47189570	0.29360858	0.16886826	0.63725843	17.81071762

Table 5
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
1	0.03111769	0.03435428	0.09279803	0.15323153	0.09346832	0.04261359	0.03001711
2	0.03509516	0.13816538	0.17451750	0.11687375	0.07826396	0.04935321	0.05484866
3	0.06562117	0.08076352	0.16740956	0.17952703	0.06085484	0.04121511	0.05182405
4	0.06517126	0.08582561	0.11033235	0.09512695	0.08154666	0.05033486	0.01915725
5	0.07793688	0.08072966	0.10732869	0.15714220	0.11792696	0.04501370	0.03599025
6	0.07585060	0.08605320	0.16565327	0.16044883	0.16821978	0.01035791	-0.00029500
7	-0.10155637	0.12415979	0.35841317	0.11760807	0.06530678	0.00767781	0.10902906
8	0.03627317	0.06245528	0.11585828	0.12926024	0.14106542	0.03602600	0.04324248
9	0.01952954	0.00399668	0.23902674	0.23394009	0.13934448	0.07640383	0.05902420
10	-0.01830002	0.25762690	0.03452605	0.21733096	0.04459127	0.07695441	0.12975943
11	0.05884086	0.15373170	0.11869195	0.16176565	0.02182462	0.06568485	0.10502301
12	0.01235211	0.17230726	0.08141107	0.16674939	0.03740025	0.02039266	0.11094840
13	-0.00444608	0.09584553	0.13453566	0.19819329	0.07434809	0.02704855	0.11198899
14	0.02489058	0.17134612	0.19475457	0.05950136	0.05832087	0.03056358	0.08973216
15	0.08236732	0.28747993	0.31235025	0.00260243	0.06295911	-0.07301595	0.07178906
16	0.11644208	0.16524757	0.14156014	0.13030822	0.08622777	0.02053839	0.02479054
17	0.05364506	0.08893608	0.23342429	0.11728463	0.16978328	0.05670414	0.05273233
18	0.03290769	0.17985448	0.17998166	0.11374621	0.13214147	0.10757679	-0.04782463
19	0.02227232	0.14338656	0.09610806	0.12864031	0.11160929	0.00396851	0.12332469
20	0.03522712	0.10894093	0.13919300	0.10783345	0.08677233	0.10112202	0.09541941
21	0.08045541	0.09044489	0.13713938	0.10415727	0.12048517	0.06659417	0.07868485
22	0.05053649	0.16402259	0.15771897	0.07327221	0.14815848	0.08242572	0.05743542
23	-0.01824534	0.17006370	0.12108318	0.01624618	0.09355890	0.13643637	0.08732430
24	0.06832238	-0.03033857	0.11820303	0.28763964	0.08263473	0.14373142	0.16264071
25	-0.04028171	0.20415884	-0.01971285	0.02108390	0.14273017	0.07010898	0.12118712

Table 5 (con.)
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
26	-0.00428549	0.15053451	0.10240003	0.04842276	0.13387820	-0.03152672	0.33145108
27	0.00351500	0.11874638	0.12800582	0.15308912	0.06437358	0.13366745	0.13724671
28	0.04905625	0.08129018	0.12446002	0.16523990	0.10567225	0.10662323	0.08208079
29	-0.02179040	0.08709843	0.21725048	0.10034018	0.10056366	0.08631938	0.09188720
30	0.04764481	0.19298383	0.07620734	0.22529102	0.06857431	0.14097801	0.06647658
31	-0.08117887	0.04791469	0.09425671	0.17696357	0.00586074	0.17675666	0.08376536
32	-0.04579640	0.10458069	0.13810341	0.16073576	0.12390547	0.16662791	0.04861894
33	0.03808216	0.14121766	0.13740345	0.11315370	0.08457348	0.12490026	0.10351769
34	0.08983760	0.21226443	0.07334474	0.20281622	0.01424586	0.07760841	0.18558215
35	-0.02508425	0.10202314	0.18140048	0.15415955	0.04896182	0.11977371	0.10765588
36	0.07022445	0.27837383	0.13745171	0.14487934	0.03998958	0.04425498	0.08112254
37	0.17469826	0.17088238	0.00203749	0.24574195	-0.05102426	0.08441096	0.10055704
38	0.07541942	0.12825076	0.05434803	0.20073820	0.00834373	0.14582594	0.22419542
39	-0.06310684	0.12907385	0.07226402	0.09894185	0.05594719	0.07621099	0.31049796
40	0.11818890	0.02006854	0.19217347	0.13496767	0.09291847	0.08948610	0.00126487
41	0.00682141	0.08277558	0.23158458	0.10130475	0.06708378	0.08319718	0.12647766
42	0.10937571	0.04462835	0.32713939	0.15167063	0.05783243	0.04508162	0.15701011
43	0.14432150	0.11472917	0.13278037	0.26430588	0.17341361	0.18161288	-0.00069111
44	-0.03399181	0.04469847	0.35522119	0.12950091	0.14694741	-0.00226911	0.19435850
45	0.12789655	0.14227855	0.16592622	0.07129149	0.09936639	0.09233109	-0.01793907
46	0.11044896	0.04955823	0.15563129	0.12729529	0.07125559	0.11459625	-0.03537109
47	-0.07675883	0.24846538	0.20433623	0.12030390	0.05241289	0.07188497	0.13742514
48	0.11541247	0.02791267	0.12611919	0.18317032	0.25512255	-0.02989174	0.02366675
49	-0.07355743	-0.00161602	0.25350199	0.23095668	0.06003877	0.19066382	0.15967828
50	0.13789501	0.32096313	0.20139275	-0.12160091	-0.03490658	0.02867959	0.12904118

Table 5 (con.)
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
51	0.16301288	0.05413537	-0.01047259	0.24661714	0.01093869	0.06228117	0.20839397
52	0.02381012	0.06839195	0.22512281	0.16366895	0.13384287	0.06583369	0.01762646
53	0.18250316	0.04145494	0.16112930	0.12468166	0.14856458	0.16250630	-0.06743747
54	-0.12993039	0.06345633	0.17455468	0.06377760	0.02284386	0.18820280	0.07377658
55	-0.08230230	0.07703912	0.21476370	0.27309529	0.06009781	0.17813531	0.03817491
56	-0.04675233	-0.07417477	0.30389953	0.32593461	-0.13273471	0.18559438	0.11517558
57	0.03354242	0.18243543	0.07769500	0.05272184	0.04968316	0.28425466	0.08173884
58	0.05267480	0.20923673	0.22191273	0.16689961	0.11527464	0.20153538	-0.00278166
59	0.01969848	0.18533925	0.23357439	0.16514711	0.15125790	0.11988275	0.07032519
60	0.09589512	0.15713603	0.10621569	0.12035911	-0.00080761	0.09239132	0.12164888
61	0.16624282	0.12734131	0.22003596	0.02255149	0.09034713	-0.06183251	0.23250646
62	0.05193193	-0.02820960	0.14260647	0.06701512	0.12607880	0.06904891	-0.05976290
63	0.03936785	0.10164020	0.34469185	0.13535301	0.12838187	0.13343646	0.02867507
64	0.10618228	0.12941283	0.24102230	0.10540467	0.13018027	0.06455584	0.00284440
65	0.13948274	0.06726175	0.13699560	0.12283464	0.20969730	0.18154498	-0.03028861
66	0.08407852	0.03947646	0.22533899	0.14259657	0.15494884	0.08237876	0.01607191
67	0.05349430	0.08359323	0.41301978	0.07467987	0.14981236	0.14306031	-0.02628570
68	-0.02992972	0.06777349	0.45247350	0.04762289	0.14615664	0.08698662	0.13185113
69	0.00261230	0.01914998	0.42984796	0.14654519	0.08445098	0.20028910	0.04495475
70	0.03608934	0.14939608	0.33333694	0.01222535	0.03585467	0.15468260	0.02190988
71	-0.02284599	0.12577101	0.15964859	0.09900467	0.09604916	0.03693651	0.14413963
72	-0.04526733	0.12345092	0.29826195	0.08021848	0.12887924	0.03320930	0.06222827
73	0.07740875	0.06728146	0.43088268	0.08331645	0.17241307	0.06837730	-0.00491789
74	-0.02390759	0.05845042	0.43747277	0.08446142	0.10433012	0.10618470	0.10715673
75	0.01585816	0.09231978	0.39939830	0.03696892	0.07642276	0.12534831	0.04825867

Table 5 (con.)
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
76	0.01865144	0.11803576	0.35042759	0.05258422	0.18043301	0.12176757	0.05624735
77	-0.01665519	0.13388071	0.44502014	0.06639025	0.13864999	0.09890750	0.07721315
78	-0.02062103	0.10381896	0.31710072	0.18395585	0.09792844	0.03340015	0.16933524
79	-0.06146090	0.21428389	0.16734781	0.04271956	0.29852214	-0.02583267	0.14684490
80	-0.01446125	0.09904441	0.23300168	0.17329381	0.11374426	0.09139959	0.13596923
81	0.06171629	0.16853199	0.20199277	0.14375284	0.12530147	0.09359370	0.09313862
82	-0.02147333	0.10584773	0.17274193	0.06846481	0.16641407	0.08753458	0.07531378
83	-0.00632047	0.12375143	0.24330083	0.06373884	0.14963530	0.15524842	0.08727075
84	0.00959989	-0.05686915	0.05446055	0.19102678	0.01558799	-0.12445937	0.21932918
85	-0.02271141	0.08269807	0.19725828	0.18001139	0.05376655	-0.06661824	0.22021316
86	0.07162249	0.18705353	0.15708632	0.15287285	0.08473196	0.12532953	0.06404562
87	0.08844477	0.27505107	0.05528562	0.14069759	0.08466360	0.11664122	0.04496436
88	0.02907033	0.11110092	0.07324119	0.16279522	-0.00678362	0.10147551	0.14296831
89	0.10855611	-0.09995771	0.18640872	0.09833233	0.15318282	0.10633984	0.07127505
90	-0.01514508	0.11760924	0.03884351	0.24908617	0.00934985	0.09148413	0.24245174
91	-0.01640634	0.33794164	0.00918374	0.17864423	0.07184360	0.00321108	0.18085575
92	-0.07305526	0.20702267	-0.02780253	0.27211495	0.03426048	0.06918522	0.27054354
93	0.05597185	0.21788330	-0.05220114	0.15553963	-0.12147840	0.09253943	0.26297659
94	-0.03424887	0.23579741	-0.06413646	0.20042735	0.00969360	-0.02604719	0.23657535
95	0.01892603	0.12997299	0.06563881	0.15051055	0.04032385	-0.07365771	0.32367506
96	-0.02410388	0.05957958	0.08622419	0.22254580	0.11312165	0.08048394	0.05681796
97	0.00712964	0.14340096	-0.04838787	0.20071681	0.04642086	0.08936228	0.08814151
98	-0.03778973	0.21383307	-0.02713654	0.22457006	-0.02109930	-0.03198122	0.17065883
99	0.07654521	0.15064437	-0.06403440	0.31410308	-0.07315489	0.03930632	0.17097842
100	-0.01008346	0.18376091	-0.06267690	0.14485034	-0.00918275	0.08999831	0.32772734

Table 5 (con.)
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
101	-0.05420188	0.25305740	0.01168652	0.26117212	0.01658431	0.02866416	0.16740679
102	0.01397524	0.24488172	-0.01891129	0.18594361	-0.03840143	0.03838258	0.16921093
103	-0.03463734	0.23756235	-0.01032518	0.23094364	-0.00783453	0.07388791	0.11082562
104	-0.07236003	0.29559780	-0.01504101	0.22034884	-0.06873029	0.09027233	0.24008972
105	0.01623813	0.25874734	0.08698924	0.18104548	-0.06308047	-0.13654996	0.18497771
106	-0.03929610	0.20966419	0.02319720	0.26882758	-0.05771797	-0.01399170	0.20185827
107	-0.03002394	0.23250621	-0.07829859	0.21769986	-0.00123038	0.09437831	0.12393439
108	0.01645063	0.15582068	0.06950469	0.09227628	0.10688011	0.08633042	0.09896169
109	-0.06452057	0.28608960	-0.00473084	0.06716386	0.11488031	0.07586420	0.24865633
110	0.05625695	0.15218638	0.19263402	0.13778618	0.10100705	0.08392213	0.04451323
111	-0.03993878	0.10999622	0.09103364	0.14168241	0.17995237	0.06061666	0.04313277
112	0.00238480	0.17117645	0.15946512	0.09561696	-0.02300704	-0.00016881	-0.04019385
113	0.09540249	0.15032447	0.15583649	0.29382200	0.06600599	0.07533334	0.02443133
114	-0.00488206	0.04389048	0.19052047	0.21746846	-0.01406684	0.04482972	0.06289541
115	0.02035332	0.11907677	0.22742931	0.04512267	0.10639384	0.07528955	0.06203723
116	-0.03823583	0.12800345	-0.00788614	0.12289262	-0.05030845	0.00727556	0.16069409
117	0.04393235	0.10289271	0.10909682	0.09528821	0.11187637	0.04802996	0.09156027
118	0.25059749	0.16563994	-0.03935491	0.16765631	0.03230471	0.03463260	0.06434538
119	0.02528986	0.19856173	-0.11214160	0.04879112	0.02888959	0.05468164	0.15850771
120	-0.14601739	0.09622084	0.10102201	0.15075030	-0.01293545	-0.01901909	0.07236690
121	0.21007926	0.21146016	0.01289526	0.05857171	0.17207014	0.05028908	0.10644065
122	-0.06567883	0.09185295	-0.08023539	0.22429669	-0.01913186	0.12937982	0.02082642
123	0.01805885	0.14948036	0.00177770	0.08424878	-0.05954322	0.13485435	0.28072313
124	0.08822603	0.21519148	0.05706704	0.25620518	0.00782912	-0.00976522	0.06314679
125	0.17426557	0.18789316	-0.01359333	0.12946213	0.14640972	0.03406472	0.06816393

Table 5 (con.)
Population Beta Weights Corrected to the 1980 Youth Population by Job Family (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE
126	0.12905332	0.27618070	-0.01969206	0.11851126	0.09289626	0.11585949	0.04979553
127	0.04177842	0.14136928	0.00924912	0.09186663	0.19449930	0.16518070	0.00308527
128	-0.02818519	0.16240525	0.05939914	0.19200754	-0.11412779	0.03544210	0.20796786
129	0.02540171	0.14413184	-0.04158243	0.01945031	0.12473450	0.03436867	0.20988337
130	-0.03576839	0.24472540	0.02992377	0.19751896	0.08068166	-0.00926058	0.12045991
131	0.06251902	0.16967625	0.17155561	0.06063401	0.09438320	0.07997696	0.13420776
132	0.17920602	0.29307560	0.11380867	-0.06300697	0.05543038	-0.01590121	0.05840615
133	-0.00550328	0.03390505	0.14698945	0.23055426	0.10878517	0.04202669	-0.04061662
134	-0.02386692	0.17544545	0.02150614	0.14790525	-0.01178449	0.04836004	0.13083855
135	-0.02183500	0.05994146	0.01570538	0.15941880	-0.00968834	0.17541289	0.26386964
136	-0.02406277	0.24787522	0.00401485	0.17912832	0.10435522	0.09078772	0.29376609
137	0.01130400	0.11506933	0.08898371	0.14289632	0.10847643	0.04481766	0.15490658
138	-0.03810761	0.19837578	0.03487066	0.08353336	0.10861723	0.01386305	0.05028711
139	-0.02308644	0.19168707	0.06853030	0.26581800	0.10845585	0.00113551	0.31897282
140	0.03832520	0.14894144	0.21019280	0.27894542	0.12537841	-0.03789962	0.18563696
141	-0.03121307	0.11495556	0.30735681	0.12223235	0.08012277	0.02672047	0.19097213
142	0.13387683	-0.00318176	-0.02471205	0.18633769	0.09251938	0.12839269	0.26768146
143	-0.04866800	0.25084402	0.05619220	0.17200757	0.10551510	0.05134043	0.27315083
144	0.00608455	0.17338653	0.01794730	0.10964830	0.05044502	0.05943074	0.08065850
145	-0.04623540	0.25389322	0.07462585	0.18812202	0.04005626	-0.03582186	0.12126457
146	0.02061056	0.16609431	0.16745115	0.22098647	0.08945763	-0.04543024	0.21093874
147	0.12897188	0.19573253	0.09383382	0.26992264	0.01953549	-0.01704065	-0.04323940
148	0.01406325	0.14478880	0.07882945	0.14117425	0.10229709	0.07555128	0.13326615
149	0.16737564	0.13664634	0.08845820	0.03012560	0.11727415	0.02716275	0.12617291
150	0.01755699	0.02096595	0.13061450	0.23873151	0.11789101	0.10409429	-0.07907066

Note. *GS = General Science; AR = Arithmetic Reasoning; NO = Numerical Operations; CS = Coding Speed; AS = Auto & Shop Information; MK = Mathematical Knowledge; EI = Electronics Information; VE = Verbal.*

Table 6
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
1	0.00412003	0.00485726	0.01050635	0.01972677	0.01107532	0.00513381	0.00537352	3.19326075
2	0.00464665	0.01953485	0.01975841	0.01504613	0.00927371	0.00594576	0.00981875	4.42167246
3	0.00868835	0.01141895	0.01895367	0.02311200	0.00721086	0.00496533	0.00927730	4.38939411
4	0.00862878	0.01213466	0.01249154	0.01224648	0.00966269	0.00606402	0.00342944	3.39953728
5	0.01031896	0.01141416	0.01215147	0.02023022	0.01397349	0.00542296	0.00644281	4.20106464
6	0.01004274	0.01216684	0.01875482	0.02065591	0.01993283	0.00124786	-0.00005281	4.35841365
7	-0.01344622	0.01755463	0.04057859	0.01514066	0.00773838	0.00092497	0.01951787	4.65580276
8	0.00480262	0.00883039	0.01311717	0.01664074	0.01671523	0.00434018	0.00774107	3.80386155
9	0.00258574	0.00056508	0.02706197	0.03011705	0.01651131	0.00920465	0.01056624	5.08454739
10	-0.00242295	0.03642520	0.00390895	0.02797882	0.00528374	0.00927098	0.02322893	5.42980204
11	0.00779062	0.02173573	0.01343799	0.02082543	0.00258606	0.00791329	0.01880073	4.88071551
12	0.00163544	0.02436208	0.00921715	0.02146703	0.00443166	0.00245678	0.01986147	4.37630758
13	-0.00058867	0.01355135	0.01523177	0.02551507	0.00880971	0.00325864	0.02004775	4.50966012
14	0.00329555	0.02422619	0.02204960	0.00766010	0.00691060	0.00368211	0.01606343	4.42321861
15	0.01090556	0.04064605	0.03536347	0.00033503	0.00746020	-0.00879650	0.01285134	5.21970891
16	0.01541711	0.02336393	0.01602706	0.01677566	0.01021737	0.00247433	0.00443789	4.65944531
17	0.00710269	0.01257444	0.02642768	0.01509902	0.02011809	0.00683135	0.00943989	5.15177355
18	0.00435703	0.02542916	0.02037705	0.01464349	0.01565781	0.01296016	-0.00856134	4.46511511
19	0.00294889	0.02027306	0.01088110	0.01656093	0.01322489	0.00047810	0.02207701	4.55262110
20	0.00466413	0.01540288	0.01575906	0.01388229	0.01028189	0.01218254	0.01708154	4.69700588
21	0.01065242	0.01278777	0.01552656	0.01340903	0.01427662	0.00802284	0.01408579	4.67455026
22	0.00669111	0.02319073	0.01785652	0.00943294	0.01755571	0.00993012	0.01028182	5.00692578
23	-0.00241571	0.02404487	0.01370872	0.00209151	0.01108605	0.01643698	0.01563239	4.25243222
24	0.00904599	-0.00428949	0.01338263	0.03703024	0.00979162	0.01731584	0.02911518	5.83842541
25	-0.00533336	0.02886549	-0.00223184	0.00271430	0.01691249	0.00844628	0.02169435	3.75323729

Table 6 (con.)
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
26	-0.00056741	0.02128369	0.01159346	0.00623386	0.01586360	-0.00379814	0.05933482	5.81645903
27	0.00046539	0.01678924	0.01449248	0.01970843	0.00762780	0.01610340	0.02456927	5.24171656
28	0.00649512	0.01149341	0.01409103	0.02127270	0.01252140	0.01284529	0.01469372	4.90857817
29	-0.00288508	0.01231462	0.02459652	0.01291762	0.01191607	0.01039921	0.01644922	4.52348131
30	0.00630825	0.02728549	0.00862799	0.02900358	0.00812556	0.01698413	0.01190033	5.66641735
31	-0.01074821	0.00677453	0.01067149	0.02278199	0.00069446	0.02129451	0.01499528	3.47932895
32	-0.00606351	0.01478639	0.01563570	0.02069285	0.01468190	0.02007427	0.00870354	4.65548027
33	0.00504214	0.01996640	0.01555645	0.01456721	0.01002135	0.01504719	0.01853125	5.19263244
34	0.01189464	0.03001152	0.00830390	0.02611021	0.00168803	0.00934977	0.03322205	6.31697725
35	-0.00332119	0.01442479	0.02053768	0.01984624	0.00580162	0.01442957	0.01927205	4.78355756
36	0.00929783	0.03935856	0.01556192	0.01865152	0.00473848	0.00533156	0.01452218	5.63854435
37	0.02313032	0.02416062	0.00023068	0.03163640	-0.00604601	0.01016929	0.01800125	5.27355316
38	0.00998565	0.01813304	0.00615314	0.02584269	0.00098867	0.01756818	0.04013442	6.22361126
39	-0.00835544	0.01824942	0.00818154	0.01273760	0.00662934	0.00918141	0.05558390	5.38800608
40	0.01564839	0.00283744	0.02175737	0.01737551	0.01101017	0.01078071	0.00022643	4.18519338
41	0.00090317	0.01170342	0.02621939	0.01304180	0.00794894	0.01002307	0.02264144	4.87628821
42	0.01448151	0.00630989	0.03703785	0.01952582	0.00685272	0.00543115	0.02810722	6.20206527
43	0.01910839	0.01622126	0.01503304	0.03402629	0.02054826	0.02187956	-0.00012372	6.64134344
44	-0.00450057	0.00631980	0.04021720	0.01667173	0.01741221	-0.00027337	0.03479315	5.85831541
45	0.01693370	0.02011640	0.01878573	0.00917794	0.01177420	0.01112346	-0.00321137	4.45413460
46	0.01462361	0.00700691	0.01762016	0.01638778	0.00844327	0.01380582	-0.00633197	3.75229877
47	-0.01016299	0.03512988	0.02313440	0.01548772	0.00621055	0.00866024	0.02460121	5.42806108
48	0.01528079	0.00394650	0.01427888	0.02358103	0.03023018	-0.00360117	0.00423671	4.64049958
49	-0.00973912	-0.00022849	0.02870082	0.02973297	0.00711416	0.02296996	0.02858486	5.63409319
50	0.01825751	0.04538015	0.02280115	-0.01565469	-0.00413618	0.00345513	0.02310035	4.91627326

Table 6 (con.)
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
51	0.02158316	0.00765406	-0.00118568	0.03174907	0.00129616	0.00750324	0.03730571	5.53450515
52	0.00315250	0.00966976	0.02548781	0.02107046	0.01585941	0.00793122	0.00315541	4.54762715
53	0.02416370	0.00586121	0.01824263	0.01605130	0.01760383	0.01957772	-0.01207234	4.69565743
54	-0.01720299	0.00897193	0.01976262	0.00821062	0.00270683	0.02267347	0.01320714	3.07610791
55	-0.01089695	0.01089236	0.02431498	0.03515782	0.00712116	0.02146060	0.00683389	4.97425135
56	-0.00619008	-0.01048738	0.03440670	0.04196026	-0.01572810	0.02235923	0.02061819	4.53454948
57	0.00444107	0.02579407	0.00879642	0.00678732	0.00588709	0.03424519	0.01463251	5.27964803
58	0.00697423	0.02958344	0.02512437	0.02148637	0.01365921	0.02427970	-0.00049796	6.33471560
59	0.00260811	0.02620464	0.02644467	0.02126076	0.01792297	0.01444271	0.01258929	6.39631755
60	0.01269666	0.02221706	0.01202546	0.01549483	-0.00009570	0.01113072	0.02177701	4.99289462
61	0.02201081	0.01800446	0.02491189	0.00290324	0.01070548	-0.00744919	0.04162222	5.95438965
62	0.00687587	-0.00398848	0.01614553	0.00862741	0.01493943	0.00831857	-0.01069848	2.12527991
63	0.00521237	0.01437065	0.03902510	0.01742512	0.01521232	0.01607557	0.00513328	5.92894993
64	0.01405870	0.01829735	0.02728790	0.01356962	0.01542542	0.00777728	0.00050919	5.10618711
65	0.01846773	0.00950997	0.01551028	0.01581352	0.02484761	0.02187138	-0.00542213	5.29302610
66	0.01113213	0.00558147	0.02551228	0.01835764	0.01836032	0.00992447	0.00287712	4.83461145
67	0.00708273	0.01181903	0.04676100	0.00961416	0.01775168	0.01723499	-0.00470554	5.57849032
68	-0.00396274	0.00958232	0.05122785	0.00613089	0.01731851	0.01047959	0.02360337	6.06450930
69	0.00034587	0.00270757	0.04866624	0.01886598	0.01000683	0.02412955	0.00804759	5.94614633
70	0.00477828	0.02112273	0.03773952	0.00157387	0.00424852	0.01863517	0.00392220	4.85467175
71	-0.00302484	0.01778244	0.01807499	0.01274569	0.01138113	0.00444987	0.02580320	4.60207659
72	-0.00599346	0.01745441	0.03376843	0.01032719	0.01527126	0.00400084	0.01113981	4.55095715
73	0.01024904	0.00951275	0.04878339	0.01072602	0.02042970	0.00823766	-0.00088038	5.66418101
74	-0.00316540	0.00826415	0.04952950	0.01087342	0.01236237	0.01279245	0.01918270	5.81169888
75	0.00209965	0.01305286	0.04521881	0.00475932	0.00905554	0.01510116	0.00863904	5.17772486

Table 6 (con.)
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
76	0.00246948	0.01668877	0.03967448	0.00676960	0.02138001	0.01466978	0.01006914	5.91174242
77	-0.00220517	0.01892905	0.05038400	0.00854697	0.01642902	0.01191574	0.01382234	6.23562152
78	-0.00273026	0.01467869	0.03590130	0.02368216	0.01160381	0.00402384	0.03031360	6.19570245
79	-0.00813752	0.03029705	0.01894667	0.00549964	0.03537272	-0.00311216	0.02628749	5.58468803
80	-0.00191469	0.01400363	0.02637983	0.02230955	0.01347787	0.01101124	0.02434058	5.77497893
81	0.00817133	0.02382830	0.02286908	0.01850650	0.01484732	0.01127557	0.01667324	6.11478286
82	-0.00284310	0.01496554	0.01955738	0.00881404	0.01971886	0.01054561	0.01348232	4.45501405
83	-0.00083684	0.01749690	0.02754587	0.00820563	0.01773070	0.01870334	0.01562280	5.51830175
84	0.00127104	-0.00804058	0.00616588	0.02459246	0.00184706	-0.01499407	0.03926329	2.63378699
85	-0.00300703	0.01169247	0.02233306	0.02317436	0.00637095	-0.00802574	0.03942153	4.84767052
86	0.00948293	0.02644702	0.01778490	0.01968059	0.01004012	0.01509890	0.01146515	5.77613915
87	0.01171022	0.03888876	0.00625929	0.01811317	0.01003202	0.01405219	0.00804931	5.61437226
88	0.00384896	0.01570827	0.00829217	0.02095798	-0.00080381	0.01222512	0.02559352	4.49433630
89	0.01437300	-0.01413276	0.02110470	0.01265914	0.01815106	0.01281114	0.01275933	4.10396057
90	-0.00200523	0.01662847	0.00439776	0.03206693	0.00110789	0.01102142	0.04340258	5.58451636
91	-0.00217222	0.04778070	0.00103976	0.02299835	0.00851295	0.00038685	0.03237595	5.82197255
92	-0.00967263	0.02927040	-0.00314773	0.03503161	0.00405962	0.00833499	0.04843144	5.88292749
93	0.00741076	0.03080596	-0.00591007	0.02002391	-0.01439431	0.01114856	0.04707684	5.01650480
94	-0.00453460	0.03333879	-0.00726136	0.02580267	0.00114862	-0.00313800	0.04235061	4.59222747
95	0.00250584	0.01837655	0.00743145	0.01937647	0.00477808	-0.00887381	0.05794280	5.34595766
96	-0.00319139	0.00842380	0.00976207	0.02865017	0.01340410	0.00969619	0.01017129	4.03655129
97	0.00094398	0.02027509	-0.00547835	0.02583994	0.00550054	0.01076580	0.01577868	3.84472549
98	-0.00500342	0.03023331	-0.00307233	0.02891077	-0.00250012	-0.00385289	0.03055055	3.93079631
99	0.01013470	0.02129922	-0.00724981	0.04043709	-0.00866833	0.00473537	0.03060776	4.74362512
100	-0.00133507	0.02598148	-0.00709611	0.01864778	-0.00108809	0.01084242	0.05866822	5.48583083

Table 6 (con.)
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
101	-0.00717641	0.03577913	0.00132312	0.03362285	0.00196512	0.00345328	0.02996838	5.17575022
102	0.00185034	0.03462319	-0.00214109	0.02393806	-0.00455029	0.00462409	0.03029135	4.63002176
103	-0.00458604	0.03358833	-0.00116899	0.02973129	-0.00092833	0.00890154	0.01983948	4.45597625
104	-0.00958058	0.04179381	-0.00170290	0.02836733	-0.00814404	0.01087543	0.04297974	5.46568701
105	0.00214995	0.03658362	0.00984869	0.02330749	-0.00747458	-0.01645067	0.03311385	4.24582595
106	-0.00520286	0.02964388	0.00262633	0.03460840	-0.00683916	-0.00168563	0.03613572	4.66183218
107	-0.00397522	0.03287345	-0.00886476	0.02802631	-0.00014579	0.01137010	0.02218616	4.24787571
108	0.00217809	0.02203108	0.00786914	0.01187949	0.01266452	0.01040054	0.01771566	4.46015022
109	-0.00854262	0.04044947	-0.00053561	0.00864656	0.01361249	0.00913963	0.04451330	5.65465000
110	0.00744851	0.02151724	0.02180951	0.01773836	0.01196861	0.01011040	0.00796855	5.18409234
111	-0.00528796	0.01555208	0.01030659	0.01823995	0.02132306	0.00730271	0.00772143	3.96423355
112	0.00031575	0.02420220	0.01805422	0.01230956	-0.00272617	-0.00002034	-0.00719532	2.35244354
113	0.01263144	0.02125399	0.01764339	0.03782614	0.00782123	0.00907568	0.00437358	5.78748367
114	-0.00064639	0.00620556	0.02157022	0.02799652	-0.00166682	0.00540080	0.01125924	3.67089117
115	0.00269481	0.01683596	0.02574894	0.00580902	0.01260690	0.00907040	0.01110561	4.43137212
116	-0.00506248	0.01809808	-0.00089285	0.01582099	-0.00596119	0.00087651	0.02876671	2.69929867
117	0.00581671	0.01454773	0.01235165	0.01226724	0.01325654	0.00578634	0.01639069	4.23709562
118	0.03317950	0.02341940	-0.00445566	0.02158379	0.00382787	0.00417231	0.01151881	4.86579973
119	0.00334842	0.02807413	-0.01269637	0.00628129	0.00342321	0.00658770	0.02837531	3.32249235
120	-0.01933293	0.01360442	0.01143744	0.01940734	-0.00153276	-0.00229130	0.01295478	1.79748273
121	0.02781482	0.02989781	0.00145997	0.00754042	0.02038907	0.00605851	0.01905451	5.90360260
122	-0.00869598	0.01298685	-0.00908404	0.02887557	-0.00226699	0.01558686	0.00372825	2.12166885
123	0.00239102	0.02113464	0.00020127	0.01084604	-0.00705544	0.01624639	0.05025374	4.92859492
124	0.01168126	0.03042537	0.00646098	0.03298342	0.00092769	-0.00117645	0.01130424	4.83442940
125	0.02307303	0.02656573	-0.00153900	0.01666673	0.01734849	0.00410390	0.01220239	5.16485992

Table 6 (con.)
Operational Weights and Constants for Computing Army Aptitude (AA) Scores by AA Composite (150 Job Families)

Job Family	GS	AR	AS	MK	MC	EI	VE	Constant
126	0.01708686	0.03904848	-0.00222948	0.01525694	0.01100754	0.01395801	0.00891416	5.39724540
127	0.00553153	0.01998784	0.00104716	0.01182675	0.02304676	0.01989992	0.00055231	4.30854310
128	-0.00373176	0.02296206	0.00672501	0.02471872	-0.01352332	0.00426984	0.03722944	4.10599495
129	0.00336323	0.02037843	-0.00470785	0.00250400	0.01478014	0.00414052	0.03757234	4.11942662
130	-0.00473579	0.03460109	0.00338789	0.02542825	0.00956019	-0.00111566	0.02156417	4.65396429
131	0.00827762	0.02399009	0.01942307	0.00780592	0.01118373	0.00963511	0.02402525	5.49983696
132	0.02372716	0.04143720	0.01288512	-0.00811141	0.00656810	-0.00191568	0.01045560	4.48234616
133	-0.00072864	0.00479375	0.01664176	0.02968116	0.01289026	0.00506311	-0.00727100	3.20218662
134	-0.00316002	0.02480578	0.00243487	0.01904107	-0.00139638	0.00582611	0.02342211	3.71436230
135	-0.00289099	0.00847497	0.00177812	0.02052330	-0.00114800	0.02113263	0.04723671	4.98572339
136	-0.00318595	0.03504644	0.00045455	0.02306067	0.01236534	0.01093753	0.05258863	6.90052405
137	0.00149667	0.01626936	0.01007450	0.01839623	0.01285367	0.00539935	0.02773065	4.85470037
138	-0.00504551	0.02804784	0.00394796	0.01075394	0.01287036	0.00167013	0.00900216	3.22663717
139	-0.00305668	0.02710214	0.00775882	0.03422096	0.01285123	0.00013680	0.05710103	7.15596826
140	0.00507432	0.02105845	0.02379747	0.03591096	0.01485644	-0.00456590	0.03323186	6.80402470
141	-0.00413266	0.01625327	0.03479812	0.01573598	0.00949397	0.00321911	0.03418694	5.78606288
142	0.01772550	-0.00044986	-0.00279783	0.02398880	0.01096288	0.01546793	0.04791909	5.91939106
143	-0.00644372	0.03546619	0.00636193	0.02214396	0.01250278	0.00618517	0.04889819	6.58394973
144	0.00080560	0.02451467	0.00203195	0.01411593	0.00597737	0.00715984	0.01443911	3.62146741
145	-0.00612164	0.03589731	0.00844894	0.02421851	0.00474638	-0.00431559	0.02170822	4.43842446
146	0.00272887	0.02348365	0.01895837	0.02844942	0.01060008	-0.00547314	0.03776127	6.12989849
147	0.01707608	0.02767412	0.01062362	0.03474938	0.00231481	-0.00205295	-0.00774051	4.30584981
148	0.00186200	0.02047131	0.00892486	0.01817453	0.01212147	0.00910194	0.02385668	4.97044193
149	0.02216080	0.01932007	0.01001500	0.00387832	0.01389614	0.00327240	0.02258688	5.01324135
150	0.00232457	0.00296432	0.01478783	0.03073389	0.01396923	0.01254062	-0.01415486	3.30546649

Note. *Operational weights and intercepts are simply unstandardized b-weights and intercepts. Unlike the 9 and 17 composites, no additional transformation is required for operational usage.*

GS = General Science; AR = Arithmetic Reasoning; NO = Numerical Operations; CS = Coding Speed; AS = Auto & Shop Information; MK = Mathematical Knowledge; EI = Electronics Information; VE = Verbal.